

# Computer Science Department

## TECHNICAL REPORT

CONCURRENCY CONTROL  
PERFORMANCE EVALUATION

(A Methodology and an Application to  
Two Phase Locking)

Oded Shmueli\*

Paul Spirakis\*\*

November 1982

Technical Report No. 54

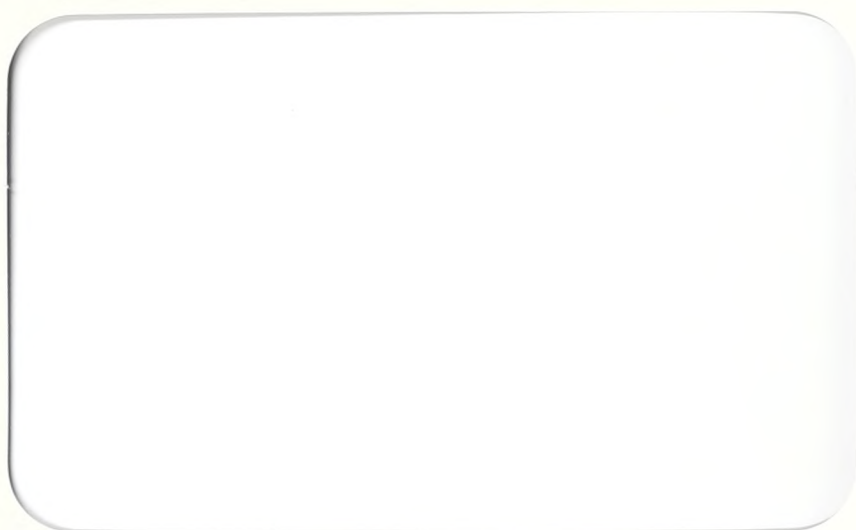
NEW YORK UNIVERSITY



Department of Computer Science  
Courant Institute of Mathematical Sciences

251 MERCER STREET, NEW YORK, N.Y. 10012

CSD TR-54  
C. Z



CONCURRENCY CONTROL  
PERFORMANCE EVALUATION

(A Methodology and an Application to  
Two Phase Locking)

Oded Shmueli\*

Paul Spirakis\*\*

November 1982

Technical Report No. 54

\* Hebrew University, Jerusalem, Israel

\*\* Courant Institute of Mathematical Sciences,  
New York University

This work was supported in part by the NSF grant NSF-MC579-21024  
and the ONR Contract N00014-80-C-0674.

The simulations for this paper were done during the summer of 1982,  
when both authors were at Harvard University.

## ABSTRACT

Concurrency Control Mechanisms (CCMs) in multi-user database systems are assigned the task of regulating the concurrent access to shared data so that basic integrity constraints of the database are preserved. Although the theory of CCM correctness has achieved a degree of mathematical maturity, the theory of quantitative performance evaluation of the various CCMs is still in its infancy. This stems from the enormous number of CCMs, the complexity of the algorithms, the lack of a precise description in some standard terminology and different (and sometimes incomparable) assumptions employed about the underlying environment.

In this paper we propose a methodology for analyzing the performance of CCMs. The methodology is based on paradigms of the physical sciences. Controlled experiments lead to hypotheses about system behavior and then a model is proposed, trying to explain the experimental data. The model is then refined by comparing its predicted results to results of new experiments. We apply this methodology to *dynamic two-phase locking* (2PL).

The experimental tools used are detailed simulation programs. Several interesting trends of behavior were observed. For example, the mean deadlock rate was observed to be logarithmic in the ratio of transactions in the system divided by total number of database items.

We also propose a model for explaining the experimental results, using the theory of the occupancy problems and Markov processes. The model captures the basic trends of the system behavior with a high degree of accuracy. Also, several variations of dynamic 2PL are compared and tested.

## 1. INTRODUCTION

Concurrency Control Mechanisms (CCMs) in multi-user database systems are assigned the task of regulating the concurrent access to shared data. A concurrency control mechanism operates correctly if it presents a consistent and coherent view to each user; one way to express that mathematically is by the theory of *serializability* [BG, 80].

In addition to correctness, good *performance* determines the usefulness of a CCM. The word performance (applied to CCMs) is a vague term and has been quantified differently by various authors. While the studies of correctness of CCMs achieved a degree of mathematical maturity, the theory of CCM performance is still in its infancy.

The reason for this stems from the following:

- The existence of an enormous number of CCMs, which tends to spread the research effort.
- The difficulty in capturing the logical characterization of CCMs by traditional queuing theoretic modelling. (The CCMs seem to require an analysis of queuing systems with simultaneous possession of many servers by a single customer, or of systems where service time distributions of any server depend on the state of another server. The above problems are not known to have a closed form solution for queuing networks of general topology.)
- The incomparability of assumptions stated by various authors (and also the believability of these assumptions, difficult to test, due to lack of commercial implementations of many of the CCMs).

- Lack of clarity of what exactly each researcher did, especially in cases of simulation-based studies.

In short, CCM performance analysis is in need of a methodology. This paper is an attempt to provide a coherent methodology for CCM study. We apply this methodology in studying dynamic two-phase locking (2PL) which is one of the few basic techniques for database concurrency control.\*

Our suggestion is to follow the paradigms of physics. An oversimplified version of this paradigm is as follows:

- A phenomenon is observed (e.g., hot gas expands; deadlocks in a database increase with number of transactions).

- Experiments are performed in a *controlled fashion* and at various ranges to obtain reliable data. (I.e., an apparatus (or a program) is carefully constructed and tested. For the gas example, this could be an apparatus measuring gas pressure as a function of temperature, for various gases and a wide range of temperatures. For a CCM this could be a simulation program which replicates, in detail, the function of the 2PL method.) Experiments should be *reproducible* by other researchers.

- Laws are conjectured in order to explain the observations. These laws are combined to a theory; a theory usually employs a model of the restricted universe of the experiments, trying to capture the fundamental trends of behavior of the system. (E.g., gas molecules are perfectly

---

\* As [BG, 80] showed, it is possible to analyze every concurrency control problem into two major subproblems, namely, read-write and write-write synchronization. Every CCM usually includes a submethod for solving each of these subproblems. Current research indicates that only a few subalgorithms are possible. Most of them are variations of two basic techniques: two-phase locking (2PL) and timestamp ordering (TO).

elastic balls, moving in the container at random directions. Transactions behave as balls, thrown randomly at a set of urns which represent the data items, in a sophisticated occupancy problem.)

- The theory predicts results of future experiments. These experiments are then performed and the theory is refined (correction-prediction method).

The *effectiveness* of a theory is measured by its ability to explain observed phenomena and to predict results of future experiments. When a theory disagrees with new experiments, when it is usually "patched" to handle exceptions, this is a good sign that a new theory is needed. Sometimes the new theory may incorporate the old as a practical projection to a restricted universe (e.g., classical mechanics and relativity).

We shall treat CCMs as physical systems. Intuitively, this is meaningful when there are large quantities of users and data and a rich variety of transaction requests to make any brute force approach absurd. The CCM should be *nice*, i.e., smooth, nonabrupt, slowly changing. Our results show that 2PL is indeed a nice CCM. We will carefully carry out the physics paradigm to 2PL [ES, G, L, T, 76]. We first define the observed system, called the *core system*. Variations of the core will correspond to variations of the 2PL method.

We concentrate here on *dynamic* methods, where locks are requested over the lifetime of a transaction. Most implementations of 2PL are dynamic. By contrast, most previous research concentrated on analysis of *static* 2PL methods (except [LN, 81], [ShS, 81]).

First we describe the experiments and tabulate results. Various trends in the observation data are then studied. A simple theory is

proposed to explain trends, and its closeness to observations is tested.

The following basic trends were observed in a large number of controlled experiments:

- The (mean) number of locked items in the database is almost linear in the number of transactions for small ratios of number of transactions over number of items. It becomes a slowly growing function of this ratio for medium values of the ratio (close to a logarithmic function). Then it stabilizes.

- The (mean) number of deadlocks within a time interval is logarithmic in the ratio of number of transactions to the number of items in the database.

- Sequential and parallel implementations of 2PL behave almost the same (for uniform data access).

- The number of deadlocks per unit time and the number of locked items have small oscillations (timewise) around their mean values.

- The rate of immediately granted lock requests is fairly stable over time.

Many other (less important) trends were also discovered. Our theory successfully explains the behavior of the mean number of locked items and the mean deadlock rate. It is based on simple combinatorial models (balls and urns) treated in a nonstandard way, and on simple Markov process theory. It predicts values with less than 15% error from actual observations.

We considered mainly *closed* database systems. (I.e., the number of transactions is kept constant over time. A completion triggers an arrival.) All simulation programs were written in Pascal and the simulations ran in a Unix/VAX environment.

Our effort is only a start. Many experimentally observed "patterns" of 2PL behavior are still to be explained. The surprising fact is that a theory to explain 2PL behavior seems within reach.

The paper is organized as follows. Section 2 surveys past work. Section 3 defines the core model and its variations; it also describes the simulation programs. Section 4 discusses the experimental results over all models. In Section 5 we propose a theory to explain the core behavior (by using a balls and urns model). Section 6 deals with variations of the core; a Markov process is proposed to explain the observed deadlock rate behavior. Section 7 outlines future work.

We present tables and plots depicting simulation observations, in Appendices II and III. Appendix I contains proofs of some claims done in the Analysis.

## 2. PAST WORK

Garcia-Molina [GM, 79] compared variants of centralized 2PL to the majority consensus algorithm of [Th, 78], mostly based on simulation studies. He concluded that centralized 2PL outperforms the Thomas's algorithm under all tested conditions. His assumptions included pre-declaration of readsets and writesets and a fully redundant database. The analysis was limited to optimistic situations in which run-time conflict never occurred. Ries [R, 79] analyzed by simulation various forms of centralized 2PL and variations of distributed 2PL. For all variations, static locking was assumed and the simulations were set up in a way that no deadlocks occurred and run-time conflicts were very rare. Consequently, the main result was that all methods had similar behavior. One of the early analytic approaches to performance evaluation of concurrency control schemes was by Gelenbe and Sevcik [GeS, 78]. Their analysis was mainly on timestamp ordering methods. However, their assumptions about the update rules are different from the current implementations. They defined measures of "coherence" (how much the values of various copies of the data agree) and "promptness" (how up-to-date the data are). Their results were based on assumptions about transmission delays and state that coherence improves rather quickly and promptness drops slowly under all methods considered. Menasce and Nakanishi [MN, 79] examined "conflict-oriented" methods analytically and locking by doing simulations. The analysis was mostly done on "certification methods." (A transaction has three phases: a read phase, a computation phase, and

a test and update phase. It can be restarted only at the end of the last phase. After the transaction has visited all the places it wants, its timestamp is compared against timestamps obtained during the read phase.) Unfortunately, their assumptions were not very clear and the model analysis seemed to be oversimplified. D. Potier and P. Leblanc [DL, 80] analyzed the case of centralized 2PL with static locking (a transaction is allowed to enter the system only if all its requested locks are available). The probability that a new transaction is granted its locks, given  $k$  other transactions already active, was found. A general limitation of all these analyses on 2PL is that only static locking is examined. Although this simplifies the analysis with respect to time dependence of state changes of the system, the issue of deadlocks and restarts will be avoided under static locking. In order to be able to quantify the degradation on performance due to restarts or deadlocks, it is important to study dynamic locking. In addition, since most real systems use dynamic locking, the primary objective of our analysis is to focus on the performance of dynamic concurrency control systems including the discussion of the rate of deadlocks.

Lin and Nolte [LN, 81] performed simulations for the case of dynamic locking, and A. Shum and P. Spirakis [ShS, 81] provided an analysis of dynamic 2PL which gave worst case bounds for any 2PL method. They considered open systems and provided an upper bound to the mean deadlock rate, proportional to the mean number of transactions in the system. The results in this paper subsume the previous results of [ShS, 81].

### 3. THE CORE MODEL AND VARIATIONS

#### 3.1. Two-Phase Locking

The database is viewed as a set of *items* (files, records, etc.). Users are modelled via *transactions*. A transaction is a sequence of operations. We consider only *read* and *write* operations. A transaction maps a consistent database state into another consistent database state. Hence, if transactions are performed serially, they all see consistent states and the final state is consistent.

When transactions operate concurrently, we need to impose restrictions on the order of events so that the execution sequence will be consistent and will lead to a consistent final state. One way to achieve this is by effectively keeping the execution equivalent to a serial one (*serializable* executions). The two-phase locking method (2PL) synchronizes read and write operations by explicitly detecting and preventing conflicts.

It regulates data access as follows. Before reading data item *x*, a transaction must own a *read lock* on *x*. Before writing into *x*, it must own a *write lock* on *x*. Different transactions cannot hold both a readlock and a writelock on an item *x*. Also, when a transaction surrenders ownership of a lock, it may not obtain any additional locks. Thus, transactions operate in a two-phase manner: one in which locks are acquired and another in which they are released.

The 2PL controller may *block* a transaction by causing it to wait for an unavailable lock. Due to this blocking, deadlocks may result.

Deadlocks could either be *prevented* or *detected* and *resolved*. Deadlock prevention is a "cautious" strategy which may lead to a large number of transaction restarts. Deadlock detection uses the *wait-for graph*. This is a directed graph whose nodes represent transactions; an edge from node *i* to node *j* means that transaction *i* waits for transaction *j* to release some locks. To resolve deadlocks when they actually occur, some transactions are restarted in order to break the deadlock cycle. We shall treat a deadlock detection 2PL implementation which was used in our simulations (the core system and its variants).

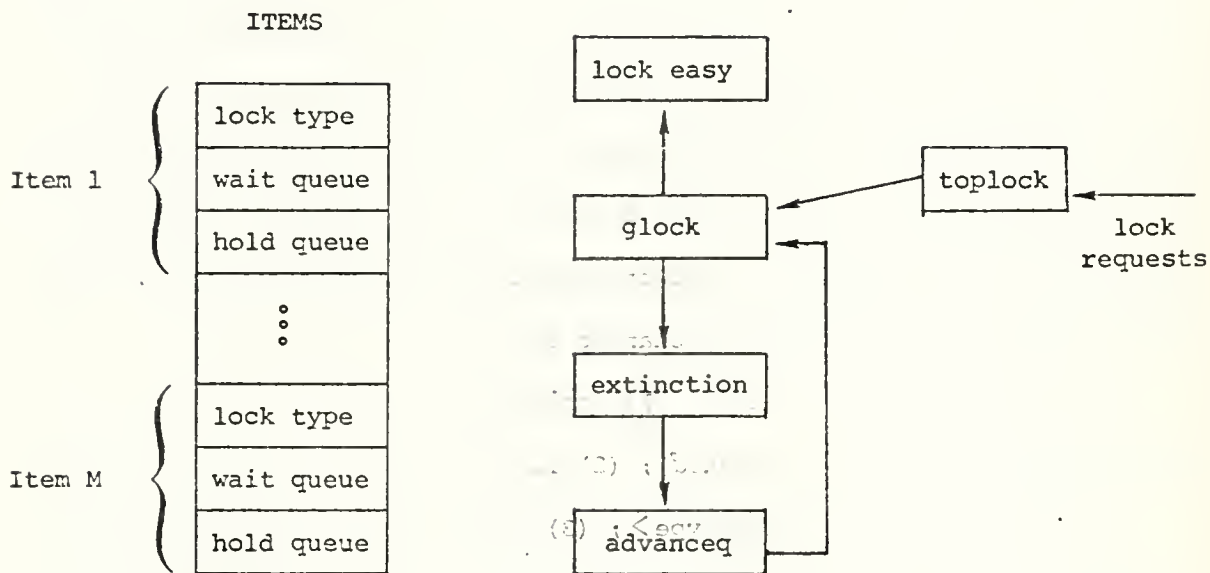
### 3.2. The Core System

In the interest of experimental reproducibility, we shall describe in detail the system we simulated. The *core system* is a particular dynamic 2PL implementation. The system is *closed*, i.e., the number of transactions (*N*) is held constant at all times. A transaction's completion is considered to trigger an immediate arrival of another transaction. We assume that no transaction tries to have more than one lock on a particular item. Figure 3.1 describes the major program modules of the core system implementation.

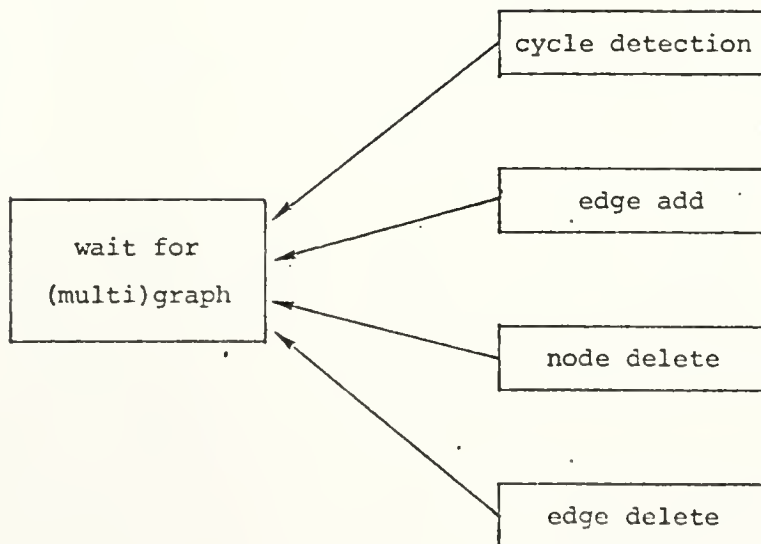
Figure 3.1(b) shows the purpose and interaction of the procedures which maintain the dynamically changing wait-for graph (wfg). These procedures are used repeatedly by the components of the system of Figure 3.1(a) which implement the locking policy. An edge in the wfg is considered to be labeled with an item number, i.e., an item on which one transaction is waiting for the other. Note that this may actually result in a multi-graph in case of a parallel implementation of 2PL. The wfg is

Figure 3.1. Core System Simulation Program Structure

(a)



(b)



maintained acyclic at all times by the deadlock resolution mechanism.

Figure 3.1(a) lists procedures in boxes. An arrow from box A to box B means that procedure A may call procedure B. The cycle (glock, extinction, advanceq) is of special interest. It indicates that a call from toplock to glock may result in "extincting" a transaction (due to deadlock). This may trigger other transactions to be advanced in their waiting queues requesting more locks, thereby (possibly) causing other deadlocks (*secondary deadlock* effect). Thus, one lock request "from outside" may induce substantial changes in the data structures.

A data item has 3 entries: (1) *locktype* (with possible values *unlocked*, *readlocked*, *writelocked*); (2) *waitq*: a (wait) queue of entries of the form  $\langle \text{transaction}, \text{locktype} \rangle$ ; (3) *holdq*: a set of transactions which hold locks on that item.

We now detail the functionality of the various simulation procedures. Denote a lock request by transaction  $j$  on item  $i$  in mode  $k$  by  $Lijk$ .

Toplock: Receives lock requests from the input stream. Calls glock ( $Lijk$ ).

Lockeasy: Called with  $Lijk$  as argument by glock. It returns 'granted' or 'postponed'. If  $i$  is unlocked then its locktype is changed to  $k$ ,  $j$  is put on  $i$ 's hold queue and 'granted' is returned. If  $k = \text{readlock}$  and  $i$ 's locktype = readlock and  $i$ 's waitqueue is empty, then  $j$  is added to  $i$ 's holdqueue, and lockeasy returns 'granted'. Otherwise, 'postponed' is returned.

Glock: It is called with  $Lijk$  and returns 'granted', 'postponed' or 'aborted'. It operates in four stages.

### Stage 1

If lockeasy (Lijk) = 'granted' then glock responds 'granted' and proceeds to stage 4.

### Stage 2

Edges (with label  $i$ ) from  $j$  to each  $t \in i$ 's holdg are added, one at a time, to indicate that  $t$  blocks  $j$ . After each edge is added, a test for cycles in the wfg is performed; in case of a cycle, one of  $j$  and  $t$  is selected at random to be restarted. (All associated edges are deleted from the wfg.) If all edges were added and there were no cycles or  $j$  is extincted, then proceed to stage 3. Else, stage 2 is repeated.

### Stage 3

3.1. If  $j$  is extincted then respond 'aborted' and go to stage 4.

3.2. If lockeasy (Lijk) = 'granted' then respond 'granted' and remove edges from the wfg from  $j$  with label  $i$ ; go to stage 4. (The reason lockeasy is called again is that changes in stage 2 might "enable" Lijk to proceed.)

3.3. Set response to 'postponed' and enqueue  $j$  on  $i$ 's wait queue.

### Stage 4

If  $j$  holds *cut* number of locks (*cut* is a parameter of the system) then extinction( $j$ ) is performed. We then consider  $j$  to be *completed*. When  $\text{cut} = \infty$ , no completions ever take place.

Extinction: It handles the extinction (due to deadlock or completion) of active (i.e., not-everywhere blocked) transactions. It is called with transaction  $j$  as argument. All edges in the wfg associated with  $j$  are

deleted. All items are scanned,  $j$  is eliminated from all wait and hold queues. Finally, for all items  $i$  in which  $j$  either held locks or was waiting,  $advanceq(i)$  is performed, since  $j$ 's extinction may enable other transactions to proceed.

Advanceq: Advances transaction in the wait queues. It is called with item  $i$ . Transactions are copied from  $i$ 's wait queue to a local queue (scratchq). Edges labelled by  $i$  are removed from the wfg and  $i$ 's wait queue is now empty. One by one, a pair  $\langle j, k \rangle$  is dequeued from scratchq. If  $j$  is not extincted,  $glock(L_{ijk})$  is performed.

### 3.3. Variations of the Core System

We considered four versions of the core system:

#### Standard Deterministic (SD)

Once a transaction is blocked (on some wait queue), no lock requests involving this transaction are generated until it is no longer on a wait queue. That is, a nonblocked transaction "moves" sequentially from item to item.

#### Parallel Deterministic (PD)

New requests may be generated involving a blocked transaction, i.e., transactions now can "split" and operate in parallel.

#### Standard Probabilistic (SP)

Same as SD except that cut is used in a probabilistic sense in  $glock$ . At stage 4, completion happens with probability  $1/cut$ . Also, all locks are considered to be writelocks.

#### Parallel Probabilistic (PP)

Same as PD, with completions happening with probability  $1/cut$ , in each new item access.

## 4. RESULTS

### 4.1. The Experiment

The input stream of lock requests (of the form  $L_{ijk}$ ) is randomly generated as follows: A nonblocked transaction  $j$  ( $1 \leq j \leq N$ ) is randomly selected out of the  $n$  nonblocked (i.e., active) transactions ( $n \leq N$ ). An item on which the transaction neither holds a lock nor waits for one is then (uniformly) randomly selected. The lock mode (readlock or write-lock) is then selected randomly as well. Thus, a lock request  $L_{ijk}$  is generated and  $toplock(L_{ijk})$  is invoked.

This process is repeated for *totlocks* number of times. *Totlocks* is the observation interval, usually in the range of 5,000-20,000 requests for ranges of total numbers of transactions ( $N$ ) and items ( $M$ ) not exceeding 500. Some cases of  $M=1000$ ,  $N \leq M$  and *totlocks* = 30,000 were also tested.

At all times the simulation program is keeping the following information.

$ni \triangleq$  number of locked database items.

$nl \triangleq$  number of locks held by all transactions.

$nw \triangleq$  sum of lengths of all waiting queues at items.

$nl_j \triangleq$  number of locks held by transaction  $j$ .

$nw_j \triangleq$  number of times transaction  $j$  appears on wait queues.

At each interval (*repperiod*) the program reports  $ni$ ,  $nl$ ,  $nw$ . Within a *repperiod* number of lock requests, the system also keeps a count of

$ns \stackrel{\Delta}{=} \text{number of successful (granted) lock requests.}$

$nd \stackrel{\Delta}{=} \text{number of deadlocks.}$

$nc \stackrel{\Delta}{=} \text{number of completions.}$

These quantities are re-initialized to 0 every *repperiod* lock requests.

In Table A.1 we present the output from one experiment. The quantities  $la$ ,  $wa$ ,  $mxla$  and  $nact$  denote (respectively) the average number of locks held by an aborted transaction, the average number of wait queue entries associated with an aborted transaction, the maximum number of locks held by an aborted transaction and the average (within *repperiod*) number of nonblocked (i.e., active) transactions, which hold at least one lock.

In Figure A.1 we plot the behavior of the quantities  $ni$ ,  $nd$ ,  $ns$  as time (i.e., lock requests) progresses. This figure is a typical one. We observe that  $ns$  is fairly stable and that  $nd$  and  $ni$  oscillate about some average value.

Table 4.1 summarizes the experimental results for SD and PD. We observed the following variations for PD (for some of the parameters).

- More items are locked.
- More deadlocks are observed.
- More waiting transactions are observed.
- Less successes ( $ns$ ).
- $mxla$  is slightly higher.

In the Appendix, we present averages for the observed quantities in tabular form. Each experiment lasted from 5,000-20,000 lock requests. For each value of cut, 27 experiments were carried out. In all our tables, transactions (N) are the horizontal axis and items (M) are the vertical axis.

Tables and diagrams (Figures 4.2 and 4.3) for the standard deterministic (SD) case appear in the Appendix.

#### 4.2. Parallel Deterministic Case

Tables and diagrams for PD appear in the Appendix.

#### 4.3. Summary for SP and PP

The results for the standard probabilistic (SP) case are very similar to those for SD. (See Appendix for tables.) For PP the results are of the same general nature, although there are some logical problems (e.g., do we probabilistically terminate even when some locks are still waited for by a transaction?). See the Appendix for more details.

#### 4.4. General Trends (in All Systems)

- Measured quantities quickly arrive to a steady-state level (oscillation around some clearly defined mean).

- $n_i$  varies linearly with N for small N, then it varies slowly (almost logarithmically) with  $N/M$  and finally stabilizes (to M) for large N.

- $n_d$  (and hence the mean deadlock rate) is a very slow function of  $N/M$  (almost logarithmic).

- ns (and hence the rate of successful requests) is a very slow (logarithmic) function of  $M/N$ . It has the smallest oscillation around its mean value among observed quantities.

- The parallel and sequential data access modes behave the same (sequential data access is slightly more efficient) for uniform data access.

- Very small mean number of active transactions.

## 5. A MODEL

In this section we present a model for explaining the behavior of  $n_i$  in our simulations. A model usually serves to decrease the level of detail in the treatment of a system, while still capturing its basic laws and functions. Hence, in designing *any* model for the core system, we face the problem of choosing the relevant "variables" of the system to keep in the model. Following the physics paradigm, we overcome the problem of over-information by ignoring many of the details of the actual "logic" of the system and try to first find a model to explain the observed quantities  $n_i$ ,  $n_d$  and  $n_s$ .

The experiments showed that  $n_i$  approaches a mean value fairly quickly, and then deadlocks, completions (if any) and restarts cause  $n_i$  to oscillate around the mean with oscillations of a relatively small magnitude. This is possibly due to the fact that the rate of deadlocks is a very slow function of  $N/M$  and that only a few (on the average) number of active transactions are in the system, even for small values of cut.

Therefore, we shall ignore deadlocks in our initial modelling effort. In Section 6, we provide a model which explains the observed deadlock rate. We utilize a nonstandard occupancy problem (for a treatment of the basic problem, see [Fe, 66]). Balls are randomly thrown into a set of urns. We may think of balls as transactions and of urns as data items. To estimate the mean number of items locked (deadlocks aside), we perform process BU (Figure 5.1).

Figure 5.1. Process BU

1. Start with a set of  $N$  balls and  $M$  empty urns (i.e.,  $N$  active transactions and all database items unlocked). Let  $n = N$ .
2. If  $n = 0$  then stop.
3. Generate a random integer  $i$ ,  $1 \leq i \leq M$ .
4.  $n \leftarrow n - 1$  |\*I.e., throw a ball\*|.
5.  $urn_i \leftarrow urn_i + 1$  |\*Add ball to  $i^{th}$  urn\*|.
6. If  $urn_i = 1$  then  $n \leftarrow n + 1$  |\*A transaction succeeded in locking an unlocked item. It can now remain an active one\*|.
7. goto 2.

Intuitively, process BU corresponds to the SD model with  $cut = \infty$  and without deadlock resolution. When BU stops (it is easy to see that it always stops after at most  $N+M$  throws), we have a deadlocked system. As deadlock resolution only introduces relatively small perturbations in  $ni$  (by the experiments), we expect BU to provide the mean value of  $ni$  (i.e., the mean number of urns hit).

Theorem 5.1. The mean number of items locked (i.e., urns occupied) after BU stops is given by the following formula:

$$ni = M \left( 1 - \left( 1 - \frac{1}{M} \right)^t \right) \quad (*)$$

where  $t$  is the (mean) time for the process to stop, and is given by the fixed point equation

$$t = N + M \left( 1 - \left( 1 - \frac{1}{M} \right)^t \right) \quad (**)$$

Theorem 5.2. Equation (\*\*) has a unique positive solution.

Proofs of Theorems 5.1 and 5.2 are given in Appendix I.

respectively.

The values of  $n_i$  from equations (\*) and (\*\*), respectively, agree to a surprising degree (error less than 15%) with the results of the system experiments, for  $\text{cut} \geq 100$ . See Table 5.1.

BU has a natural extension to SP (or to SD with  $\text{cut} < \infty$ ). The process is modified to capture completion with probability  $\theta (= 1/\text{cut})$ . The new process is described in Figure 5.2.

Figure 5.2. Process BU ( $\theta$ )

1. Start with  $N$  balls,  $M$  urns,  $n \leftarrow N$
2. If  $n = 0$  then stop
3. Generate a random integer  $i$ ,  $1 \leq i \leq M$
4.  $n \leftarrow n - 1$
5.  $\text{urn}_i \leftarrow \text{urn}_i + 1$
6. If  $\text{urn}_i = 1$  then
  - begin
    - $x \leftarrow$  random number in  $[0,1]$
    - if  $x < \theta$  then           no-op
    - else            $n \leftarrow n + 1$
  - end
7. goto 2

Intuitively, when a transaction hits an empty item it either decides to complete (with probability  $\theta$ ) or to continue being active (with probability  $(1-\theta)$ ).

The corresponding theorems are:

Theorem 5.3. The mean number of items locked when  $BU(\theta)$  stops is given by the following formula:

$$n_i = M \left( 1 - \left( 1 - \frac{1}{M} \right)^t \right) \quad (***)$$

where  $t$  is the (mean) time for  $BU(\theta)$  to finish and is the solution of the fixed point equation

$$t = N + (1-\theta)M \left( 1 - \left( 1 - \frac{1}{M} \right)^t \right) \quad (****)$$

Note that for  $\theta = 0$  (i.e., cut =  $\infty$ ) we get Theorem 5.1. Also, we get

Theorem 5.4. Equation (\*\*\*\*) has a unique positive solution.

Proofs of Theorems 5.3 and 5.4 are given in Appendix I. Again, we have a very close agreement between  $n_i$  (from  $BU(\theta)$ ) and  $n_i$  (experimental) with less than 15% error, for any  $\theta$ . (See Appendix II, III).

## 5.2. Nonempty Urns Distribution (BU, cut = $\infty$ )

Let  $p_t(b, r)$  = probability {at time  $t$  there are  $b$  filled urns and  $r$  balls remain}. Then

$$p_t(b, r) = p_{t-1}(b, r+1) \frac{b}{M} + p_{t-1}(b-1, r) \left(1 - \frac{b-1}{M}\right)$$

with  $p_0(N, M) = 1$

$$p_t(b, r) = 0 \quad \text{for } r < 0$$

$$p_t(b, r) = 0 \quad \text{for } b > M.$$

We are interested in the distributions of (1) the time  $t^*$  such that  $r_{t^*} = 0$ , (2) the probability  $p_{t^*}(b, 0)$  for  $0 \leq b \leq M$  which we call  $P(m)$  where  $m = M - b$ .

Suppose that exactly  $m$  urns are empty at the end. Since  $M-m$  urns were occupied, exactly  $M-m$  new balls were created (in addition to the initial  $N$ ). Hence, we have a total of  $N+M-m$  balls placed in  $M-m$  urns, so that each of these urns is occupied. From classical occupancy problems (Feller, pp. 58, 92) this probability is

$$\sigma(m) = M^{N+M-m} \binom{M}{m} \sum_{v=0}^{M-m} (-1)^v \binom{M-m}{v} (M-m-v)^{N+M-m}$$

Hence  $P(m) = \lambda \cdot \sigma(m)$  where  $\lambda$  is such that

$$\sum_{m=0}^M \sigma(m) = \frac{1}{\lambda}$$

When  $N \rightarrow \infty$ , we know from sampling theory (Feller, pp. 210-211) that

$$\text{mean}(t^*/m) \simeq M \log \frac{M}{M - m + 1}.$$

## 6. RATES OF DEADLOCK AND OF SUCCESSFUL LOCK REQUESTS

We consider a model for explaining the mean deadlock rate and the mean request success rate, for the SD and SP variations of the case.

### 6.1. Theoretic Estimation of $\bar{n}_d$ and $\bar{n}_s$

We shall assume here that all locks are writelocks. Let us consider a (modified) wfg where an edge from transaction  $T$  to transaction  $T'$  exists if and only if either

- (1)  $T$  is the first transaction on the wait queue for some item  $X$  and  $T'$  currently owns a lock on  $X$  or
- (2)  $T'$  is immediately in front of  $T$  on the wait queue for some item  $X$ . Note that due to the sequential nature of the SP and SD systems, there can be at most one edge emanating from a particular transaction  $T$  at any particular point in time. A transaction  $T$  is *blocked* if there is an edge emanating from it, else it is considered to be *free*. New (or restarted) transactions enter the system as free transactions. Whenever a new edge is added to the graph, it must emanate from a previously free transaction (since they are the only ones which can generate lock requests).

At any point in time (deadlocks apart), the wfg is a forest of trees whose tree roots are the free transactions. The total number of nodes of the wfg is always  $N$  (closed system), and the number ( $n$ ) of trees may vary from 1 to  $N$ . (See also [ShS, 81].)

We can easily define a discrete time Markov process describing the system's evolution. Let the state of the system be  $(s_1, \dots, s_n, k)$  where

$n \triangleq$  number of free transactions at time  $t_k$  and  $s_i \triangleq$  size of tree  $i$  at  $t_k$ . At each discrete time, one of the free transactions is selected at random and makes a lock request (or completes with prob  $\theta$  for the SP system). If the lock is granted, the state vector  $(s_1, \dots, s_n, k+1)$  is identical in the first  $n$  components to  $(s_1, \dots, s_n, k)$ . Else, a conflict happens (an edge is formed from the selected tree root to another tree or to that root's tree). If the conflict is not a deadlock, the two trees merge into a single tree whose size equals the sum of the two sizes. If the conflict is a deadlock (when the root conflicts with its own tree), then the deadlock resolution mechanism instantaneously restarts the root which caused the deadlock and the tree is broken into  $x+1$  trees where  $x$  is the number of sons of the deadlocked root.

A completion situation is similar to a deadlock in terms of state transitions. The distribution of  $x$  is state-dependent and one can (for example) use the number of random partitions of the integer  $s_i - 1$  ( $i$  is the deadlocked tree to be broken (see [ShS, 81])). Let  $p_{ij}(t_k) =$  probability {the root of the tree  $T_i$  will conflict with tree  $T_j$  at  $t_{k+1}$ , given that the root of  $T_i$  is selected to make a request at time  $t_k$ }. Then, if  $L_j(t_k)$  is the number of items locked by all the transactions in the tree  $T_j$  at  $t_k$ , we can estimate  $p_{ij}(t_k)$  to be

$$p_{ij}(t_k) = \frac{L_j(t_k)}{M}$$

The total probability of conflict at  $t_{k+1}$  will then be

$$\alpha = \sum_{i=1}^n \sum_{j=1}^n \frac{1}{n} \cdot (1-\theta) \cdot p_{ij}(t_k)$$

$$\text{or } \alpha = \frac{1-\theta}{n} \cdot n \left( \sum_{j=1}^n L_j(t_k) \right) \frac{1}{M}$$

$$\text{or } \alpha = (1-\theta) \frac{1}{M} \cdot L(t_k) \quad (\text{Eq. 6.1})$$

where  $L(t_k)$  = number of locked items at instant  $t_k$ .

Recalling our results of Section 5 (and simulation), we may approximate  $L(t_k)$  by its mean value and get

$$\alpha \approx (1-\theta) \frac{1}{M} \cdot ni(N, M) \quad (\text{Eq. 6.2})$$

where  $ni(N, M) = M \left( 1 - \left( 1 - \frac{1}{M} \right)^{t^*} \right)$  and  $t^*$  is given by Theorems 5.1 or 5.3. Equation (6.2) gives us the mean rate of conflicts in the system.

The mean rate of successful requests will then be

$$rs = 1 - \alpha \approx 1 - (1-\theta) \left( 1 - \left( 1 - \frac{1}{M} \right)^{t^*} \right) \quad (\text{Eq. 6.3})$$

where  $t^*$  is as in Theorem 5.3 or 5.1. Note that  $(rs * \text{totlocks})$  is a theoretic estimation of the ns in the experiments. We found that  $(rs * \text{totlocks})$  is a close upper bound to ns for both SD and SP.

The total probability of deadlock at  $t_{k+1}$  is

$$\begin{aligned} pd(t_k) &= \sum_{i=1}^n \frac{1}{n} \cdot p_{ii}(t_k) \cdot (1-\theta) \\ &= \frac{1-\theta}{n} \frac{L(t_k)}{M} \end{aligned} \quad (\text{Eq. 6.4})$$

By comparing Equations 6.4 and 6.1 we conclude that

Lemma 6.1. The probability of a deadlock at a state where  $n$  free transaction exist is equal to  $\alpha/n$ , where  $\alpha$  is the probability of conflict at the same state. The average deadlock rate is equal to

$$r_d = \sum_{n=1}^N f(n) \cdot \frac{\alpha}{n}$$

where  $f(n)$  = probability that the number of free transactions in the system (steady state) is  $n$ .

We need to estimate  $f(n)$ . Let  $\text{cut} < \infty$  and consider the following Markov process of continuous time. The state variable is the number of free transactions in the system ( $n$ ). We consider the following state transitions and rates.

$n \rightarrow n-1$  with rate  $\mu_n = (1-\theta)\alpha \cdot (1-\frac{1}{n})$ , i.e. a conflict which is not a deadlock.

$n \rightarrow n+\text{cut}$  with rate  $\lambda_n = \theta + (1-\theta)\frac{\alpha}{n}$  (i.e. either a completion, or a deadlock, we use  $\theta = \frac{1}{\text{cut}}$ ).

This simple process is expected to capture the effect of deadlocks and conflicts on the mean number of free transactions. We should remark that the above process describes a *bulk arrival system* Klei, 76 with constant bulk size  $b = \text{cut}$ , with arrival rate  $\lambda_n = \theta + (1-\theta)\frac{\alpha}{n}$  and completion rate  $\mu_n = \alpha \cdot (1-\theta)(1-\frac{1}{n})$ . The above process is equivalent to an Erlangian System with  $\text{cut} = \text{number of stages}$  and with state dependent rates. The steady state density of  $n$  can be shown to be a sum of geometrics

$$(\text{Eq. 6.6}) \quad f(K) = p(n=K) = \frac{1-p}{G} \sum_{i=1}^{\text{cut}} (c_i)^{-K} \quad K = 1, 2, \dots, N$$

where  $G$  is a normalizing constant depending on  $\text{cut}$ ,  $N$ ,  $\theta$  and  $\alpha$ ,

$$p = \frac{\lambda'}{\mu'}, \quad \text{where } \lambda', \mu' \text{ are the mean}$$

arrival and departure rates and the  $c_i$ 's are constants (depending on  $\rho$  and  $\text{cut}$ ),

positive and less than 1. (Proof omitted. See [Klei,76] pp 129-130).

Equation 6.6 implies that the mean value for  $n$  is a very slowly growing function of  $N$  (bounded by  $\frac{(1-\rho)}{G} \sum_{i=1}^{\text{cut}} \frac{1}{i}$ ) much smaller than  $N$ . All our simulations validate this point (see Section 4 and the Appendix).

It is easy to remark that an  $f(n)$  which has high values only on small  $n$  gives the worst case rate of deadlock. Hence, if we replace  $\text{cut}$  with  $\text{cut}=1$  in the above Markov process, we decrease the births and we will get an  $f(n)$  concentrated on small  $n$  and hence a pessimistic  $r_d$ . The modified process with  $\text{cut} = 1$  is a generalized birth-death process with solution

$$f_m(K) = f_m(1) \frac{\lambda_1 \cdot \lambda_2 \cdot \dots \cdot \lambda_{K-1}}{\mu_2 \cdot \mu_3 \cdot \dots \cdot \mu_K}$$

and

$$f_m(1) = \frac{1}{1 + \sum_{k=2}^N \left( \frac{\lambda_1 \cdot \dots \cdot \lambda_{k-1}}{\mu_2 \cdot \dots \cdot \mu_k} \right)}$$

Hence

$$r_d \leq \sum_{n=1}^N f_m(n) \frac{\alpha}{n} \quad (\text{Eq. 6.7})$$

Formula (6.7) theoretically justifies the slow (almost logarithmic) increase of the rate of deadlocks with  $N$ . (Note  $\alpha \leq 1$ ). We have used  $(r_d * \text{totlocks})$  as a theoretic estimate of  $n_d$ . It is always an upper bound of  $n_d$  for both the SD and the SP systems.

### 6.3. Mean Number of Free Transactions and Deadlock Rate

We examine how the mean number of free transactions relates to the average rate of deadlock; we consider  $\text{cut} = \infty$  for the SP and SD variations. Let  $n_t \triangleq$  mean number of free transactions at time  $t$  ( $n_0 = N$ ) and let  $d_t$  = number of deadlocks at time  $t$ . Let  $\alpha = \alpha(N, M)$  be the probability of conflict (as in Eq. 6.2). Let us also assume that, when a deadlock happens, a number  $c$  of new free transactions is added to  $n_t$  and that the mean value of  $c$  is a function of  $N$  and  $M$  not depending on  $t$ . Then,

$$n_{t+1} = (n_t + c) \frac{\alpha}{n_t} + (n_t - 1) \alpha \cdot \left(1 - \frac{1}{n_t}\right) + n_t (1 - \alpha) \quad (\text{Eq. 6.7})$$

$$(\text{since } \text{probability}(\text{deadlock}) = \frac{\alpha}{n_t})$$

$$\text{probability}(\text{success}) = 1 - \alpha$$

$$\text{probability}(\text{conflict but no deadlock}) = \alpha \left(1 - \frac{1}{n_t}\right).$$

$$\text{Also,} \quad d_{t+1} = d_t + \frac{\alpha}{n_t} \quad (\text{Eq. 6.8})$$

$$\text{Equation 6.7 implies} \quad n_{t+1} = n_t - \alpha + \frac{\alpha}{n_t} (c+1) \quad (\text{Eq. 6.8})$$

by elementary manipulation.

$$\text{Equation 6.8 implies} \quad d_t = \alpha \sum_{i=0}^{t-1} \frac{1}{n_i} \quad (\text{Eq. 6.9})$$

The mean deadlock rate is defined to be equal to

$$r_d \triangleq \lim_{t \rightarrow \infty} \frac{d_t}{t} \quad (\text{Eq. 6.10})$$

From Eq. 6.8 we get,

$$\frac{\alpha}{n_i} = \frac{n_{i+1} - n_i}{c + 1} + \frac{\alpha}{c + 1}$$

So, by Eq. 6.9,

$$d_t = \frac{n_t - n_o}{c + 1} + \frac{\alpha \cdot t}{c + 1}$$

and

$$\frac{dt}{t} = \frac{n_t - n_o}{t \cdot (c+1)} + \frac{\alpha}{c + 1}$$

Since  $n_t \leq N$  we get,

$$r_d = \lim_{t \rightarrow \infty} \frac{d_t}{t} = \frac{\alpha}{c + 1} \quad (\text{Eq. 6.10})$$

Corollary: The mean deadlock rate (when cut =  $\infty$ ) is proportional to the ratio of the mean probability of conflict divided by the sum of one and the mean number of transactions freed in each deadlock.

## 7. CONCLUSIONS AND FUTURE WORK

We aimed at formulating a *methodology* for analyzing CCMS. We followed the traditional paradigms of experimental sciences. Experiments were performed in a controlled fashion to obtain observation data and gain intuition concerning CCMS' behavior. In those (reproducible) experiments, we discovered that 2PL behaves in a "physical" fashion.

A model was constructed and solved analytically; it successfully explained trends regarding the number of locked items in the system. The model is based on the classical occupancy problem with balls representing transactions and urns representing data items. We have also provided analytic modelling of the second order effects of deadlocks, and of the mean number of free transactions in the system.

The next step is to extend the modelling effort in order to explain more of the observations. Only once this is done, and the behavior of the dominant 2PL system "variables" is understood, will we feel ready to tackle the important issues of:

- (a) throughput,
- (b) response time,
- (c) influence of read/write mix on system behavior,
- (d) other lock request distributions,
- (e) locality of requests,
- (f) extensions to distributed databases.

# REFERENCES

- [BCMP, 75] Baskett, F., Chandy, K.M., Muntz, R.R., and Palacios, J., Open, closed and mixed networks of queues with different classes of customers, *JACM* 22, 2 (1975), 248-260.
- [BG, 78] Bernstein, P., and N. Goodman, "Approaches to concurrency control in distributed database systems," Harvard University, Center for Research in Computing Technology, TR-26-78, 1978.
- [BG, 80] Bernstein, P., and Goodman, A.. Fundamental algorithms for concurrency control in distributed database systems, CCA TR Contract No. F30603-79-0191, Cambridge, MA (1980).
- [BSW, 79] Bernstein, P., Shipman, D., and Wong, W., Formal aspects of serializability in database concurrency control, *IEEE Trans. Softw. Eng.* SE-5, 3 (1979), 203-215.
- [Ca, 79] Cayley, A., A theorem on trees, *Quart. J. Math.* 23 (1889), 376-378. Mathematical Papers, Cambridge, England, 13 (1897), 26-28.
- [DB, 78] Denning, P.J., and Buzen, J.P., The operational analysis of queuing network models, *ACM Comp. Surveys* 10, 3 (1978), 225-261.
- [ES, G, L, T, 76] Eswaran, K.P., Gray, J.N., Lorie, R.A. and Traiger, I.L., The notions of consistency and predicate laws in a database system, *Comm. ACM* 19, 11 (Nov 1976), 624-633.
- [Fe, 66] Feller, W., *An Introduction to Probability Theory and Its Applications*, Vol. I (Wiley, New York, 1966).
- [Ges, 78] Gelenbe, E., and Sevcik, K., Analysis of update synchronization for multiple copy databases, Proc. 3rd Berkeley Workshop on Distributed Databases and Computer Networks (August, 1978).
- [GM, 79] Garcia-Molina, H., Performance of Update Algorithms for Replicated Data in a Distributed Database, Ph.D. Thesis, Computer Science Dept., Stanford University (June, 1975).
- [Ha, 69] Harary, F., *Graph Theory* (Addison-Wesley Series in Mathematics, 1969).
- [Klei, 76] Kleinrock, L., *Queuing Systems*, Vol. I (Wiley, New York, 1976).
- [LN, 81] Lin, K. and Nolte, J., "Performance of 2PL," CCA TR, 1981.

- [MN, 79] Menasce, D., and Nakanishi, T., Optimistic Versus Pessimistic Concurrency Control Mechanisms in Database Management Systems, Dept. de Informatica, Pontificia Universidad Catolica do Rio de Janeiro, Brasil (1979).
- [PL, 80] Potier, D., and Leblanc, P.L., Analysis of locking policies in database management systems, *CACM* 23, 10 (1980), 584-593.
- [R, 79] Ries, D., The Effect of Concurrency Control on Database Management System Performance, Ph.D. Thesis, Computer Science Dept., University of California, Berkeley (April, 1979).
- [RLS, 78] Rosenkrantz, D.J., Stearns, R.E., and Lewis, D.M., System level concurrency control for distributed database systems, *ACM Transactions on Database Systems* 3, 2 (June, 1978), 178-198.
- [ShS, 81] Shum, A. and Spirakis, P., Performance analysis of concurrency control methods in database systems, Performance '81, F.J. Kylstra (editor), North Holland Publishing Company, 1981.
- [Th, 79] Thomas, R.H., A majority consensus approach to concurrency control for multiple copy databases, *ACM TODS* 4, 2 (1979), 180-204.
- [Th, 78] Thomas, R.H., A solution to the concurrency control problem for multiple copy databases, Proc. 1978 COMPCON Conf. IEEE, New York (1978).

## Appendix I,

Mean Value Analysis of BU and BU( $\theta$ )

We begin by proving Theorem 5.1. Let  $b_{t+1}^{\Delta}$  mean number of filled urns at time  $t+1$  and  $r_{t+1}^{\Delta}$  remaining balls outside of urns at time  $t+1$  (mean). Then the following recurrence relations hold:

$$b_{t+1} = b_t \cdot \frac{b_t}{M} + (b_t + 1) \left(1 - \frac{b_t}{M}\right) \quad (\text{Eq. 1})$$

$$r_{t+1} = r_t \left(1 - \frac{b_t}{M}\right) + (r_t - 1) \frac{b_t}{M} \quad (\text{Eq. 2})$$

since, the probability that, at time  $t$ , a ball will hit an empty box, is equal to  $b_t/M$ .

$$(\text{Eq. 1}) \text{ implies } b_{t+1} = b_t \left(1 - \frac{1}{M}\right) + 1$$

$$(\text{Eq. 2}) \text{ implies } r_{t+1} = r_t - \frac{b_t}{M}$$

Since  $b_0 = 0$  and  $r_0 = N$ , we get

$$b_1 = 1 \text{ and } b_t = \left(1 - \frac{1}{M}\right)^{t-1} + \left(1 - \frac{1}{M}\right)^{t-2} + \dots + 1,$$

$$\text{implying } b_t = M \left(1 - \left(1 - \frac{1}{M}\right)^t\right) \quad (\text{Eq. 3})$$

$$\text{and } r_t = N - \frac{1}{M} \sum_{i=1}^{t-1} b_i, \text{ implying}$$

$$r_t = N - t + M \left(1 - \left(1 - \frac{1}{M}\right)^t\right) \quad (\text{Eq. 4})$$

Let  $t^*$  be the time at which BU stops. Then  $r_{t^*} = 0$  and thus

$$t^* = N + M \left(1 - \left(1 - \frac{1}{M}\right)^{t^*}\right)$$

$$\text{and } b_{t^*} = n_i = M \left(1 - \left(1 - \frac{1}{M}\right)^{t^*}\right) \text{ by Equation 3.}$$

Note that  $N < t^* < N + M$ .

For the proof of Theorem 5.3, let  $b_t, r_t$  be defined as above and let  $\theta \triangleq$  probability of completion. Again,  $b_0 = 0, r_0 = N$  and

$$b_{t+1} = b_t \cdot \frac{b_t}{M} + (b_t + 1) \left(1 - \frac{b_t}{M}\right) \quad (\text{Eq. a})$$

But now,

$$r_{t+1} = (r_t - 1) \left[ \frac{b_t}{M} + \left(1 - \frac{b_t}{M}\right) \cdot \theta \right] + r_t \cdot (1 - \theta) \left(1 - \frac{b_t}{M}\right) \quad (\text{Eq. b})$$

Again Equation (a) implies that

$$b_t = M \left(1 - \left(1 - \frac{1}{M}\right)^t\right) \quad (\text{Eq. c})$$

However, now

$$r_{t+1} = r_t - \frac{b_t}{M} (1 - \theta) - \theta$$

implying

$$r_1 = N - \theta \quad \text{and} \quad r_t = N - \frac{1 - \theta}{M} \sum_{i=1}^{t-1} b_i - t \cdot \theta$$

By using the expression for  $b_i, i < t$ , we get (by elementary calculations)

$$r_t = N - t + (1 - \theta)M \left(1 - \left(1 - \frac{1}{M}\right)^t\right)$$

leading to the following expression for  $t^*$  (such that  $r_{t^*} = 0$ ):

$$t^* = N + (1 - \theta)M \left(1 - \left(1 - \frac{1}{M}\right)^{t^*}\right) \quad (\text{Eq. d})$$

and hence by Equation c

$$n_i = M \left(1 - \left(1 - \frac{1}{M}\right)^{t^*}\right) \quad (\text{Eq. e})$$

Equations (d) and (e) prove Theorem 5.3. □

Note again that  $N < t^* < N + (1 - \theta)M$  for  $BU(\theta)$ 's termination time.

Uniqueness of  $t^* > 0$ 

Next, we prove Theorems 5.2 and 5.4. We prove that the equation

$$t = N + (1-\theta)M \left(1 - \left(1 - \frac{1}{M}\right)^t\right) \triangleq f(t)$$

$$(N, M > 0, \quad \theta \in [0, 1])$$

has a unique positive solution. This will prove both Theorems 5.4 and 5.2 ( $\theta = 0$ ).

Consider first the sequence  $t_0 = N + (1-\theta)M$  and  $t_{i+1} = f(t_i)$ . Clearly,  $t_1 < t_0$  and  $t_1 < N + (1-\theta)M$ . Assume (inductively) that  $t_i < t_{i-1}$  and  $t_i < N + (1-\theta)M$ . We have

$$t_{i+1} = N + (1-\theta)M \left[1 - \left(1 - \frac{1}{M}\right)^{t_i}\right]$$

$$t_i = N + (1-\theta)M \left[1 - \left(1 - \frac{1}{M}\right)^{t_{i-1}}\right]$$

implying

$$\frac{N + (1-\theta)M - t_{i+1}}{N + (1-\theta)M - t_i} = \left(1 - \frac{1}{M}\right)^{t_i - t_{i-1}}$$

Since  $0 < 1 - \frac{1}{M} < 1$  and  $t_i < t_{i-1}$  we get

$$\frac{N + (1-\theta)M - t_{i+1}}{N + (1-\theta)M - t_i} > 1$$

implying that

$$t_{i+1} < t_i < N + (1-\theta)M.$$

So,  $\{t_i\}$  is strictly decreasing and bounded below (by  $N$ ); hence it converges.

In fact, any sequence of  $t_0^\alpha = \alpha > 0$  and  $t_i^\alpha = f(t_{i-1}^\alpha)$  will converge (same proof), to a nonzero number. We still have to prove that all  $\{t_i^\alpha\}$  will converge to the same (nonzero) limit.

Let  $A = t_i^\alpha$ ,  $B = t_i^\beta$   $\alpha, \beta > 0$

and  $A^+ = t_{i+1}^\alpha = f(t_i^\alpha)$ ,  $B^+ = t_{i+1}^\beta = f(t_i^\beta)$ .

Let  $\Delta_i = |A - B|$

$$\begin{aligned}\Delta_{i+1} &= |A^+ - B^+| = (1-\theta)^M \left| \left(1 - \frac{1}{M}\right)^A - \left(1 - \frac{1}{M}\right)^B \right| \\ &= (1-\theta)^M \left(1 - \frac{1}{M}\right)^{\max(A,B)} \left| 1 - \left(1 - \frac{1}{M}\right)^{\Delta_i} \right| \\ &\leq (1-\theta)^M \left(1 - \frac{1}{M}\right)^{\max(A,B)} \cdot \Delta_i \cdot \frac{1}{M} \\ &\leq (1-\theta) \left(1 - \frac{1}{M}\right)^{\max(A,B)} \cdot \Delta_i\end{aligned}$$

Hence

$$\Delta_{i+1} \leq \Delta_0 (1-\theta)^i \left(1 - \frac{1}{M}\right)^{\left(\sum_{k=0}^i \max(t_k^\alpha, t_k^\beta)\right)} = g(i)$$

But the sequence  $g(i) \rightarrow 0$  as  $i \rightarrow \infty$ . Hence  $\Delta_i \rightarrow 0$  as  $i \rightarrow \infty$ . So both  $\{t_i^\alpha\}$ ,  $\{t_i^\beta\}$  converge to the same positive limit. Hence  $t = f(t)$  has a unique positive solution.  $\square$

Table A.1.

N=10 M=320 Variation=PP

Cut=100 Totlocks=5000

Report every 100 requests

| ns     | nc | nd | ni  | nl  | nw | la | wa | mxla | nact |
|--------|----|----|-----|-----|----|----|----|------|------|
| 93     | 0  | 0  | 88  | 93  | 7  | 0  | 0  | 0    | 3    |
| 89     | 0  | 5  | 92  | 97  | 4  | 18 | 0  | 27   | 6    |
| 87     | 0  | 3  | 114 | 116 | 5  | 25 | 0  | 34   | 5    |
| 85     | 0  | 5  | 94  | 97  | 8  | 22 | 0  | 43   | 2    |
| 88     | 0  | 5  | 102 | 107 | 8  | 17 | 0  | 43   | 2    |
| 86     | 0  | 5  | 83  | 85  | 8  | 23 | 0  | 34   | 2    |
| 90     | 0  | 5  | 69  | 72  | 8  | 21 | 0  | 62   | 2    |
| 86     | 0  | 5  | 76  | 78  | 7  | 18 | 0  | 48   | 3    |
| 87     | 0  | 4  | 91  | 96  | 8  | 19 | 0  | 48   | 2    |
| 86     | 0  | 5  | 76  | 79  | 7  | 22 | 0  | 40   | 3    |
| 90     | 0  | 4  | 87  | 91  | 5  | 21 | 0  | 48   | 5    |
| 88     | 0  | 3  | 63  | 68  | 6  | 39 | 0  | 56   | 3    |
| 90     | 0  | 5  | 97  | 100 | 9  | 12 | 1  | 21   | 1    |
| 86     | 0  | 7  | 61  | 61  | 4  | 19 | 0  | 51   | 6    |
| 88     | 0  | 4  | 65  | 67  | 8  | 21 | 0  | 34   | 2    |
| 84     | 0  | 4  | 88  | 95  | 7  | 17 | 0  | 28   | 3    |
| 84     | 0  | 7  | 76  | 81  | 5  | 15 | 0  | 28   | 5    |
| 87     | 0  | 5  | 95  | 100 | 5  | 15 | 0  | 36   | 5    |
| 83     | 0  | 7  | 59  | 60  | 5  | 19 | 0  | 43   | 5    |
| 89     | 0  | 3  | 86  | 91  | 9  | 20 | 1  | 45   | 1    |
| 93     | 0  | 4  | 73  | 74  | 6  | 29 | 0  | 67   | 4    |
| 86     | 0  | 4  | 69  | 70  | 8  | 24 | 0  | 31   | 2    |
| 82     | 0  | 9  | 54  | 57  | 7  | 11 | 0  | 23   | 3    |
| 88     | 0  | 4  | 61  | 62  | 6  | 23 | 0  | 33   | 4    |
| 89     | 0  | 5  | 80  | 82  | 6  | 15 | 0  | 33   | 4    |
| 85     | 0  | 5  | 92  | 93  | 8  | 16 | 0  | 35   | 2    |
| 83     | 0  | 5  | 78  | 78  | 7  | 22 | 0  | 36   | 3    |
| 88     | 0  | 4  | 84  | 87  | 7  | 21 | 0  | 42   | 3    |
| 86     | 0  | 7  | 87  | 89  | 7  | 13 | 0  | 21   | 3    |
| 85     | 0  | 5  | 89  | 89  | 9  | 18 | 0  | 29   | 1    |
| 88     | 0  | 4  | 106 | 110 | 9  | 18 | 0  | 22   | 1    |
| 89     | 0  | 6  | 86  | 89  | 5  | 19 | 0  | 39   | 5    |
| 83     | 0  | 8  | 63  | 64  | 6  | 14 | 0  | 36   | 4    |
| 87     | 0  | 5  | 69  | 71  | 3  | 18 | 0  | 67   | 6    |
| 84     | 0  | 5  | 90  | 90  | 2  | 15 | 0  | 23   | 8    |
| 83     | 0  | 5  | 130 | 134 | 9  | 8  | 0  | 11   | 1    |
| 85     | 0  | 6  | 60  | 62  | 3  | 28 | 0  | 83   | 7    |
| 88     | 0  | 3  | 98  | 100 | 8  | 18 | 0  | 24   | 2    |
| 83     | 0  | 7  | 79  | 82  | 7  | 16 | 0  | 35   | 3    |
| 84     | 0  | 4  | 87  | 88  | 9  | 22 | 0  | 37   | 1    |
| 86     | 0  | 6  | 94  | 96  | 9  | 14 | 0  | 26   | 1    |
| 89     | 0  | 5  | 80  | 83  | 5  | 22 | 0  | 53   | 5    |
| 86     | 0  | 4  | 79  | 81  | 9  | 23 | 0  | 36   | 1    |
| 87     | 0  | 6  | 128 | 130 | 9  | 7  | 0  | 15   | 1    |
| 88     | 1  | 2  | 84  | 90  | 9  | 18 | 0  | 20   | 1    |
| 87     | 0  | 6  | 83  | 87  | 5  | 16 | 0  | 27   | 5    |
| 87     | 0  | 5  | 102 | 106 | 8  | 14 | 0  | 23   | 1    |
| 87     | 0  | 7  | 67  | 68  | 8  | 18 | 0  | 69   | 2    |
| 90     | 0  | 3  | 100 | 107 | 9  | 20 | 0  | 28   | 1    |
| 87     | 0  | 6  | 79  | 79  | 8  | 19 | 0  | 45   | 2    |
| vg: 86 | 0  | 5  | 83  | 86  | 6  | 18 | 0  | 36   | 3    |

Figure A.1

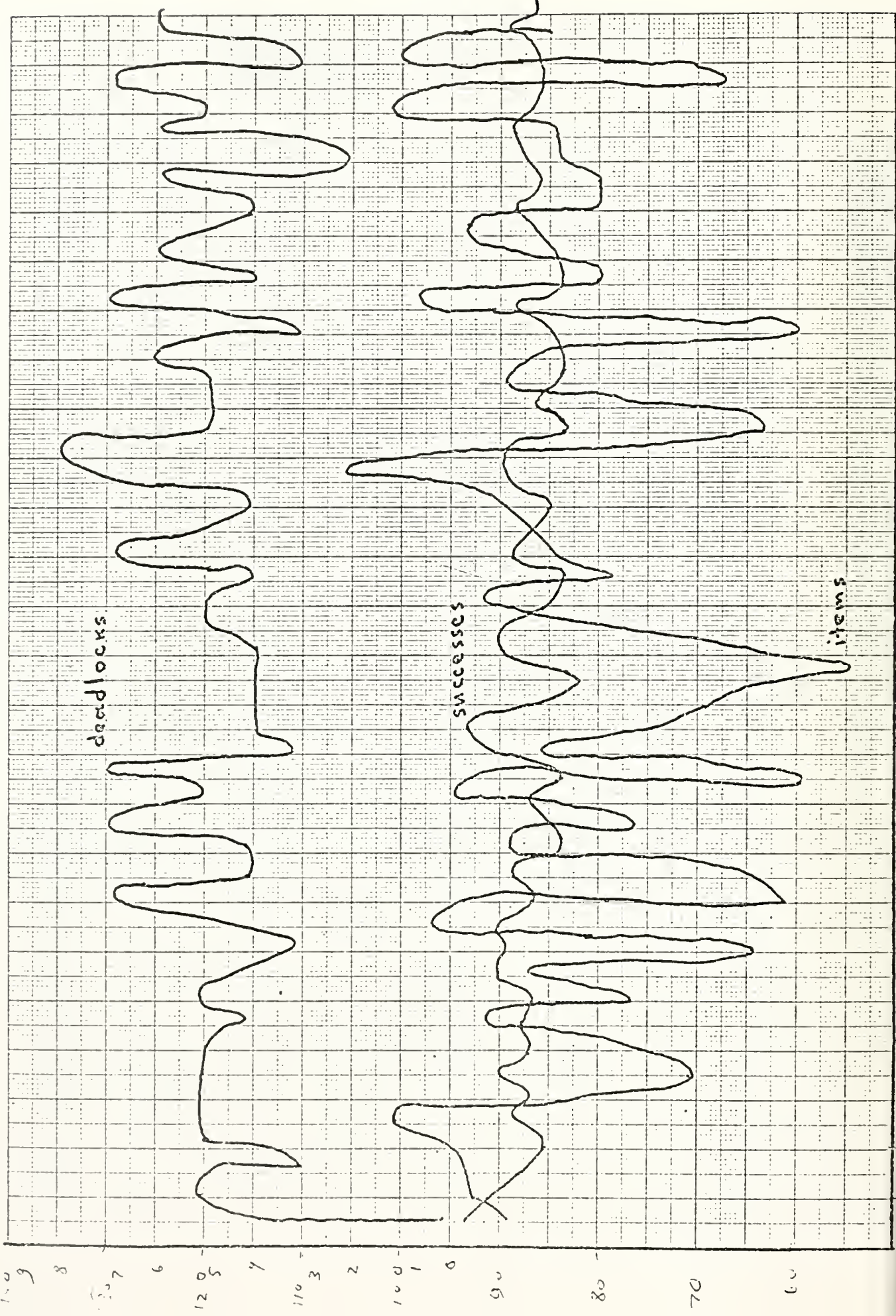


Table A.2

Example of comparison of model and simulation

(a)

|       |  | Mean Number of Urns Occupied (BU) - Theoretical |     |     |     |     |     |
|-------|--|-------------------------------------------------|-----|-----|-----|-----|-----|
| M \ N |  | 10                                              | 20  | 40  | 80  | 160 | 320 |
|       |  |                                                 |     |     |     |     |     |
| 10    |  | 8                                               | 9   | 10  | 10  | 10  | 10  |
| 20    |  | 14                                              | 17  | 19  | 20  | 20  | 20  |
| 40    |  | 22                                              | 28  | 34  | 38  | 39  | 40  |
| 80    |  | 34                                              | 44  | 56  | 67  | 76  | 79  |
| 160   |  | 50                                              | 68  | 88  | 112 | 135 | 152 |
| 320   |  | 24                                              | 101 | 135 | 177 | 224 | 269 |

(b)

System SD, cut = 100

|       |  | Mean Number of Data Items Locked - Simulation |     |     |     |     |     |
|-------|--|-----------------------------------------------|-----|-----|-----|-----|-----|
| M \ N |  | 10                                            | 20  | 40  | 80  | 160 | 320 |
|       |  |                                               |     |     |     |     |     |
| 10    |  |                                               |     |     |     |     |     |
| 20    |  | 14                                            | 16  | 18  | 19  |     |     |
| 40    |  | 22                                            | 28  | 33  | 36  | 38  |     |
| 80    |  | 37                                            | 46  | 54  | 65  | 71  | 77  |
| 160   |  | 55                                            | 68  | 90  | 109 | 128 | 141 |
| 320   |  | 83                                            | 103 | 140 | 173 | 214 | 249 |

Table A.3. SD and PD Behavior

| Cut  | 100                                       | 10                                         | 4                    |
|------|-------------------------------------------|--------------------------------------------|----------------------|
| ni   | First $\sim \log(N/M)$ ,<br>then stable M | Between logarithmic<br>and linear in (N/M) | Linear in N          |
| nd   | Logarithmic in (N/M)                      | Logarithmic in (N/M)                       | Logarithmic in (N/M) |
| nw   | Linear in N                               | Unclear                                    | Unclear              |
| ns   | Logarithmic in (N/M)                      | Logarithmic in (N/M)                       | Logarithmic in (N/M) |
| mxla | Slightly faster than<br>$\log(N/M)$       | Log (N/M)                                  | Log (N/M)            |

### APPENDIX III.

#### Simulation Results

In all tables, the horizontal axis is transactions (N) and the vertical axis is items (M);  $\theta = 1/\text{cut}$ . The averages are on 5000 - 20000 lock requests and 27 controlled experiments. The reporting period in each experiment was 100 lock requests.

A • 0 Averages for measured quantities of the standard deterministic (SD) system,

Cut = 100

| ns           |     | Transactions (N) |    |    |    |     |     |
|--------------|-----|------------------|----|----|----|-----|-----|
|              |     | 10               | 20 | 40 | 80 | 160 | 320 |
| Items<br>(M) | 20  | 51               | 36 | 21 | 12 |     |     |
|              | 40  | 61               | 48 | 33 | 22 | 12  |     |
|              | 80  | 73               | 62 | 47 | 35 | 22  | 13  |
|              | 160 | 80               | 71 | 60 | 48 | 35  | 23  |
|              | 320 | 86               | 78 | 70 | 60 | 48  | 36  |

| nc  |  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|--|----|----|----|----|-----|-----|
| 20  |  | 0  | 0  | 0  | 0  |     |     |
| 40  |  | 0  | 0  | 0  | 0  | 0   |     |
| 80  |  | 0  | 0  | 0  | 0  | 0   | 0   |
| 160 |  | 0  | 0  | 0  | 0  | 0   | 0   |
| 320 |  | 0  | 0  | 0  | 0  | 0   | 0   |

| nd  |  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|--|----|----|----|----|-----|-----|
| 20  |  | 19 | 25 | 32 | 38 |     |     |
| 40  |  | 13 | 19 | 25 | 31 | 38  |     |
| 80  |  | 9  | 13 | 18 | 24 | 31  | 37  |
| 160 |  | 6  | 9  | 13 | 18 | 23  | 30  |
| 320 |  | 4  | 7  | 9  | 13 | 17  | 24  |

| ni  |  | 10 | 20  | 40  | 80  | 160 | 320 |
|-----|--|----|-----|-----|-----|-----|-----|
| 20  |  | 15 | 16  | 18  | 19  |     |     |
| 40  |  | 29 | 28  | 32  | 36  | 38  |     |
| 80  |  | 35 | 46  | 54  | 65  | 72  | 76  |
| 160 |  | 53 | 69  | 89  | 110 | 128 | 142 |
| 320 |  | 77 | 105 | 138 | 175 | 213 | 254 |

| nl  |  | 10 | 20  | 40  | 80  | 160 | 320 |
|-----|--|----|-----|-----|-----|-----|-----|
| 20  |  | 17 | 20  | 23  | 26  |     |     |
| 40  |  | 31 | 32  | 39  | 47  | 52  |     |
| 80  |  | 37 | 52  | 62  | 79  | 92  | 104 |
| 160 |  | 55 | 74  | 99  | 127 | 156 | 183 |
| 320 |  | 80 | 109 | 147 | 190 | 243 | 309 |

Cut = 100 (cont'd.)

| nw  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 6  | 16 | 36 | 75 |     |     |
| 40  | 6  | 15 | 36 | 75 | 154 |     |
| 80  | 6  | 16 | 35 | 75 | 153 | 308 |
| 160 | 6  | 15 | 35 | 74 | 153 | 306 |
| 320 | 6  | 15 | 35 | 74 | 151 | 309 |

| la  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 3  | 2  | 1  | 1  |     |     |
| 40  | 4  | 3  | 2  | 1  | 0   |     |
| 80  | 8  | 5  | 3  | 2  | 1   | 0   |
| 160 | 13 | 8  | 5  | 3  | 2   | 1   |
| 320 | 19 | 12 | 8  | 5  | 3   | 2   |

| wa  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 0  | 0  | 0  | 0  |     |     |
| 40  | 0  | 0  | 0  | 0  | 0   |     |
| 80  | 0  | 0  | 0  | 0  | 0   | 0   |
| 160 | 0  | 0  | 0  | 0  | 0   | 0   |
| 320 | 0  | 0  | 0  | 0  | 0   | 0   |

| m×la | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 11 | 7  | 5  | 3  |     |     |
| 40   | 12 | 11 | 7  | 5  | 4   |     |
| 80   | 22 | 17 | 10 | 8  | 5   | 3   |
| 160  | 27 | 20 | 15 | 10 | 8   | 5   |
| 320  | 38 | 28 | 19 | 15 | 10  | 8   |

| nact | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 2  | 2  | 2  | 2  |     |     |
| 40   | 2  | 3  | 2  | 2  | 2   |     |
| 80   | 3  | 3  | 3  | 3  | 3   | 4   |
| 160  | 2  | 3  | 4  | 4  | 4   | 6   |
| 320  | 3  | 4  | 4  | 4  | 5   | 5   |

Cut = 16

| ns  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 50 | 35 | 22 | 12 |     |     |
| 40  | 64 | 49 | 35 | 22 | 14  |     |
| 80  | 74 | 61 | 49 | 35 | 22  | 13  |
| 160 | 81 | 71 | 61 | 47 | 35  | 23  |
| 320 | 88 | 80 | 71 | 60 | 49  | 36  |

| nc  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 0  | 0  | 0  | 0  |     |     |
| 40  | 0  | 0  | 0  | 0  | 0   |     |
| 80  | 1  | 0  | 0  | 0  | 0   | 0   |
| 160 | 3  | 1  | 0  | 0  | 0   | 0   |
| 320 | 4  | 3  | 1  | 0  | 0   | 0   |

| nd  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 19 | 25 | 31 | 38 |     |     |
| 40  | 13 | 18 | 24 | 31 | 37  |     |
| 80  | 8  | 13 | 18 | 24 | 31  | 37  |
| 160 | 4  | 8  | 12 | 18 | 24  | 30  |
| 320 | 1  | 4  | 8  | 12 | 17  | 24  |

| ni  | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|-----|-----|-----|-----|
| 20  | 14 | 16 | 18  | 19  |     |     |
| 40  | 23 | 28 | 33  | 36  | 38  |     |
| 80  | 34 | 44 | 56  | 64  | 72  | 76  |
| 160 | 44 | 67 | 87  | 111 | 129 | 143 |
| 320 | 60 | 96 | 133 | 178 | 216 | 252 |

| nl  | 10 | 20  | 40  | 80  | 160 | 320 |
|-----|----|-----|-----|-----|-----|-----|
| 20  | 16 | 21  | 24  | 27  |     |     |
| 40  | 25 | 32  | 40  | 47  | 52  |     |
| 80  | 36 | 48  | 65  | 78  | 91  | 102 |
| 160 | 46 | 72  | 95  | 128 | 158 | 184 |
| 320 | 61 | 100 | 141 | 197 | 249 | 308 |

Cut = 16 (cont'd.)

| nw  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 6  | 16 | 35 | 75 |     |     |
| 40  | 6  | 16 | 35 | 75 | 154 |     |
| 80  | 6  | 15 | 36 | 74 | 153 | 308 |
| 160 | 6  | 15 | 35 | 74 | 153 | 306 |
| 320 | 4  | 14 | 34 | 73 | 152 | 309 |

| la  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 3  | 2  | 1  | 1  |     |     |
| 40  | 5  | 3  | 2  | 1  | 1   |     |
| 80  | 7  | 5  | 3  | 2  | 1   | 0   |
| 160 | 8  | 7  | 5  | 3  | 2   | 1   |
| 320 | 9  | 8  | 6  | 5  | 3   | 2   |

| wa  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 0  | 0  | 0  | 0  |     |     |
| 40  | 0  | 0  | 0  | 0  | 0   |     |
| 80  | 0  | 0  | 0  | 0  | 0   | 0   |
| 160 | 0  | 0  | 0  | 0  | 0   | 0   |
| 320 | 0  | 0  | 0  | 0  | 0   | 0   |

| m×la | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 9  | 8  | 5  | 4  |     |     |
| 40   | 11 | 10 | 8  | 5  | 4   |     |
| 80   | 12 | 12 | 10 | 8  | 5   | 3   |
| 160  | 12 | 12 | 12 | 10 | 8   | 5   |
| 320  | 10 | 12 | 12 | 12 | 10  | 7   |

| nact | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 2  | 2  | 2  | 2  |     |     |
| 40   | 2  | 3  | 3  | 2  | 2   |     |
| 80   | 3  | 3  | 2  | 3  | 3   | 4   |
| 160  | 3  | 3  | 3  | 4  | 4   | 6   |
| 320  | 4  | 4  | 5  | 5  | 5   | 6   |

Cut = 12

| ris | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 50 | 34 | 21 | 10 |     |     |
| 40  | 64 | 50 | 35 | 22 | 13  |     |
| 80  | 75 | 63 | 49 | 35 | 23  | 14  |
| 160 | 83 | 72 | 61 | 48 | 35  | 23  |
| 320 | 90 | 82 | 71 | 60 | 49  | 37  |

| nc  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 0  | 0  | 0  | 0  |     |     |
| 40  | 1  | 0  | 0  | 0  | 0   |     |
| 80  | 4  | 1  | 0  | 0  | 0   | 0   |
| 160 | 6  | 3  | 1  | 0  | 0   | 0   |
| 320 | 7  | 6  | 3  | 1  | 0   | 0   |

| nd  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 18 | 25 | 32 | 39 |     |     |
| 40  | 12 | 18 | 24 | 31 | 38  |     |
| 80  | 6  | 11 | 17 | 24 | 30  | 37  |
| 160 | 3  | 7  | 12 | 17 | 24  | 30  |
| 320 | 0  | 2  | 7  | 12 | 17  | 24  |

| ni  | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|-----|-----|-----|-----|
| 20  | 14 | 16 | 18  | 19  |     |     |
| 40  | 22 | 28 | 32  | 36  | 38  |     |
| 80  | 32 | 43 | 53  | 65  | 71  | 75  |
| 160 | 41 | 63 | 87  | 108 | 129 | 142 |
| 320 | 50 | 82 | 127 | 168 | 217 | 253 |

| nl  | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|-----|-----|-----|-----|
| 20  | 16 | 20 | 24  | 27  |     |     |
| 40  | 24 | 33 | 39  | 47  | 52  |     |
| 80  | 34 | 47 | 61  | 78  | 90  | 101 |
| 160 | 42 | 67 | 96  | 123 | 156 | 185 |
| 320 | 50 | 85 | 135 | 185 | 250 | 306 |

Cut = 12 (cont'd.)

| nw  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 6  | 16 | 35 | 76 |     |     |
| 40  | 6  | 16 | 35 | 75 | 154 |     |
| 80  | 5  | 15 | 35 | 75 | 153 | 309 |
| 160 | 4  | 14 | 34 | 74 | 152 | 306 |
| 320 | 3  | 13 | 34 | 73 | 151 | 309 |

| la  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 3  | 2  | 1  | 0  |     |     |
| 40  | 4  | 3  | 2  | 1  | 0   |     |
| 80  | 5  | 4  | 3  | 2  | 1   | 1   |
| 160 | 7  | 5  | 4  | 3  | 2   | 1   |
| 320 | 4  | 6  | 5  | 4  | 3   | 2   |

| wa  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 0  | 0  | 0  | 0  |     |     |
| 40  | 0  | 0  | 0  | 0  | 0   |     |
| 80  | 0  | 0  | 0  | 0  | 0   | 0   |
| 160 | 0  | 0  | 0  | 0  | 0   | 0   |
| 320 | 0  | 0  | 0  | 0  | 0   | 0   |

| m×la | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 9  | 7  | 5  | 3  |     |     |
| 40   | 10 | 9  | 8  | 5  | 4   |     |
| 80   | 9  | 9  | 9  | 7  | 5   | 4   |
| 160  | 9  | 9  | 9  | 9  | 7   | 5   |
| 320  | 4  | 8  | 9  | 9  | 8   | 7   |

| nact | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 2  | 2  | 2  | 2  |     |     |
| 40   | 2  | 2  | 2  | 2  | 2   |     |
| 80   | 3  | 3  | 3  | 3  | 3   | 4   |
| 160  | 4  | 4  | 4  | 4  | 4   | 6   |
| 320  | 5  | 6  | 5  | 5  | 6   | 6   |

Cut = 8

ns

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 52 | 34 | 21 | 9  |     |     |
| 40  | 65 | 50 | 35 | 21 | 12  |     |
| 80  | 78 | 64 | 49 | 36 | 23  | 13  |
| 160 | 86 | 77 | 63 | 48 | 36  | 23  |
| 320 | 93 | 85 | 75 | 63 | 49  | 36  |

nc

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 2  | 0  | 0  | 0  |     |     |
| 40  | 6  | 2  | 0  | 0  | 0   |     |
| 80  | 9  | 6  | 2  | 0  | 0   | 0   |
| 160 | 11 | 9  | 6  | 2  | 0   | 0   |
| 320 | 12 | 11 | 9  | 6  | 2   | 0   |

nd

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 16 | 25 | 31 | 39 |     |     |
| 40  | 9  | 16 | 24 | 32 | 38  |     |
| 80  | 3  | 9  | 16 | 23 | 30  | 37  |
| 160 | 1  | 3  | 8  | 16 | 23  | 30  |
| 320 | 0  | 0  | 3  | 8  | 16  | 22  |

ni

|     | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|-----|-----|-----|-----|
| 20  | 14 | 17 | 18  | 19  |     |     |
| 40  | 20 | 27 | 32  | 36  | 38  |     |
| 80  | 26 | 40 | 54  | 64  | 72  | 75  |
| 160 | 30 | 51 | 82  | 107 | 127 | 143 |
| 320 | 35 | 63 | 106 | 158 | 209 | 251 |

nl

|     | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|-----|-----|-----|-----|
| 20  | 16 | 21 | 24  | 27  |     |     |
| 40  | 22 | 31 | 40  | 46  | 51  |     |
| 80  | 27 | 44 | 62  | 77  | 95  | 102 |
| 160 | 31 | 54 | 88  | 123 | 153 | 188 |
| 320 | 36 | 65 | 111 | 172 | 238 | 305 |

Cut = 8 (cont'd.)

| nw  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 6  | 16 | 36 | 75 |     |     |
| 40  | 6  | 15 | 35 | 75 | 154 |     |
| 80  | 4  | 14 | 35 | 75 | 153 | 309 |
| 160 | 3  | 11 | 33 | 73 | 153 | 306 |
| 320 | 2  | 9  | 32 | 72 | 150 | 303 |

| la  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 3  | 2  | 1  | 1  |     |     |
| 40  | 3  | 3  | 2  | 1  | 0   |     |
| 80  | 4  | 3  | 3  | 2  | 1   | 1   |
| 160 | 4  | 4  | 3  | 3  | 2   | 1   |
| 320 | 0  | 2  | 4  | 3  | 3   | 2   |

| wa  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 0  | 0  | 0  | 0  |     |     |
| 40  | 0  | 0  | 0  | 0  | 0   |     |
| 80  | 0  | 0  | 0  | 0  | 0   | 0   |
| 160 | 0  | 0  | 0  | 0  | 0   | 0   |
| 320 | 0  | 0  | 0  | 0  | 0   | 0   |

| m:la | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 6  | 6  | 5  | 3  |     |     |
| 40   | 6  | 6  | 6  | 5  | 4   |     |
| 80   | 6  | 6  | 6  | 6  | 5   | 3   |
| 160  | 4  | 5  | 6  | 6  | 6   | 5   |
| 320  | 0  | 3  | 5  | 6  | 6   | 5   |

| nact | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 2  | 2  | 2  | 2  |     |     |
| 40   | 3  | 3  | 2  | 2  | 2   |     |
| 80   | 4  | 4  | 3  | 3  | 3   | 3   |
| 160  | 5  | 6  | 5  | 5  | 4   | 5   |
| 320  | 6  | 9  | 6  | 6  | 6   | 8   |

Cut = 4

| ns  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 59 | 39 | 22 | 12 |     |     |
| 40  | 76 | 57 | 37 | 25 | 13  |     |
| 80  | 88 | 75 | 57 | 39 | 25  | 14  |
| 160 | 93 | 86 | 74 | 56 | 40  | 26  |
| 320 | 96 | 92 | 86 | 73 | 57  | 39  |

| nc  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 18 | 11 | 5  | 1  |     |     |
| 40  | 23 | 18 | 11 | 5  | 1   |     |
| 80  | 24 | 23 | 19 | 12 | 5   | 1   |
| 160 | 24 | 24 | 23 | 19 | 12  | 5   |
| 320 | 24 | 24 | 24 | 23 | 19  | 11  |

| nd  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 7  | 18 | 28 | 38 |     |     |
| 40  | 1  | 7  | 17 | 27 | 37  |     |
| 80  | 0  | 1  | 6  | 16 | 27  | 36  |
| 160 | 0  | 0  | 1  | 6  | 15  | 26  |
| 320 | 0  | 0  | 0  | 1  | 5   | 15  |

| ni  | 10 | 20 | 40 | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|
| 20  | 11 | 15 | 17 | 19  |     |     |
| 40  | 13 | 22 | 30 | 34  | 38  |     |
| 80  | 14 | 27 | 43 | 59  | 69  | 75  |
| 160 | 14 | 29 | 54 | 86  | 115 | 136 |
| 320 | 14 | 28 | 58 | 105 | 171 | 233 |

| nl  | 10 | 20 | 40 | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|
| 20  | 12 | 18 | 22 | 26  |     |     |
| 40  | 14 | 24 | 35 | 43  | 52  |     |
| 80  | 14 | 28 | 47 | 70  | 87  | 99  |
| 160 | 14 | 29 | 57 | 93  | 134 | 171 |
| 320 | 14 | 28 | 60 | 110 | 185 | 269 |

Cut = 4 (cont'd.)

| nw  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 4  | 15 | 34 | 75 |     |     |
| 40  | 2  | 13 | 33 | 75 | 154 |     |
| 80  | 1  | 8  | 30 | 73 | 153 | 309 |
| 160 | 1  | 5  | 23 | 68 | 151 | 306 |
| 320 | 0  | 2  | 11 | 54 | 144 | 301 |

| la  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 1  | 1  | 1  | 1  |     |     |
| 40  | 1  | 1  | 1  | 1  | 0   |     |
| 80  | 0  | 1  | 1  | 1  | 1   | 0   |
| 160 | 0  | 0  | 1  | 1  | 1   | 1   |
| 320 | 0  | 0  | 0  | 1  | 1   | 1   |

| wa  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 0  | 0  | 0  | 0  |     |     |
| 40  | 0  | 0  | 0  | 0  | 0   |     |
| 80  | 0  | 0  | 0  | 0  | 0   | 0   |
| 160 | 0  | 0  | 0  | 0  | 0   | 0   |
| 320 | 0  | 0  | 0  | 0  | 0   | 0   |

| m×la | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 2  | 3  | 2  | 2  |     |     |
| 40   | 2  | 2  | 3  | 2  | 2   |     |
| 80   | 0  | 2  | 2  | 2  | 2   | 2   |
| 160  | 0  | 0  | 1  | 2  | 2   | 2   |
| 320  | 0  | 0  | 0  | 1  | 2   | 2   |

| nact | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 3  | 3  | 3  | 2  |     |     |
| 40   | 5  | 4  | 4  | 2  | 2   |     |
| 80   | 6  | 8  | 6  | 4  | 3   | 4   |
| 160  | 6  | 10 | 12 | 8  | 5   | 6   |
| 320  | 6  | 12 | 21 | 19 | 10  | 10  |

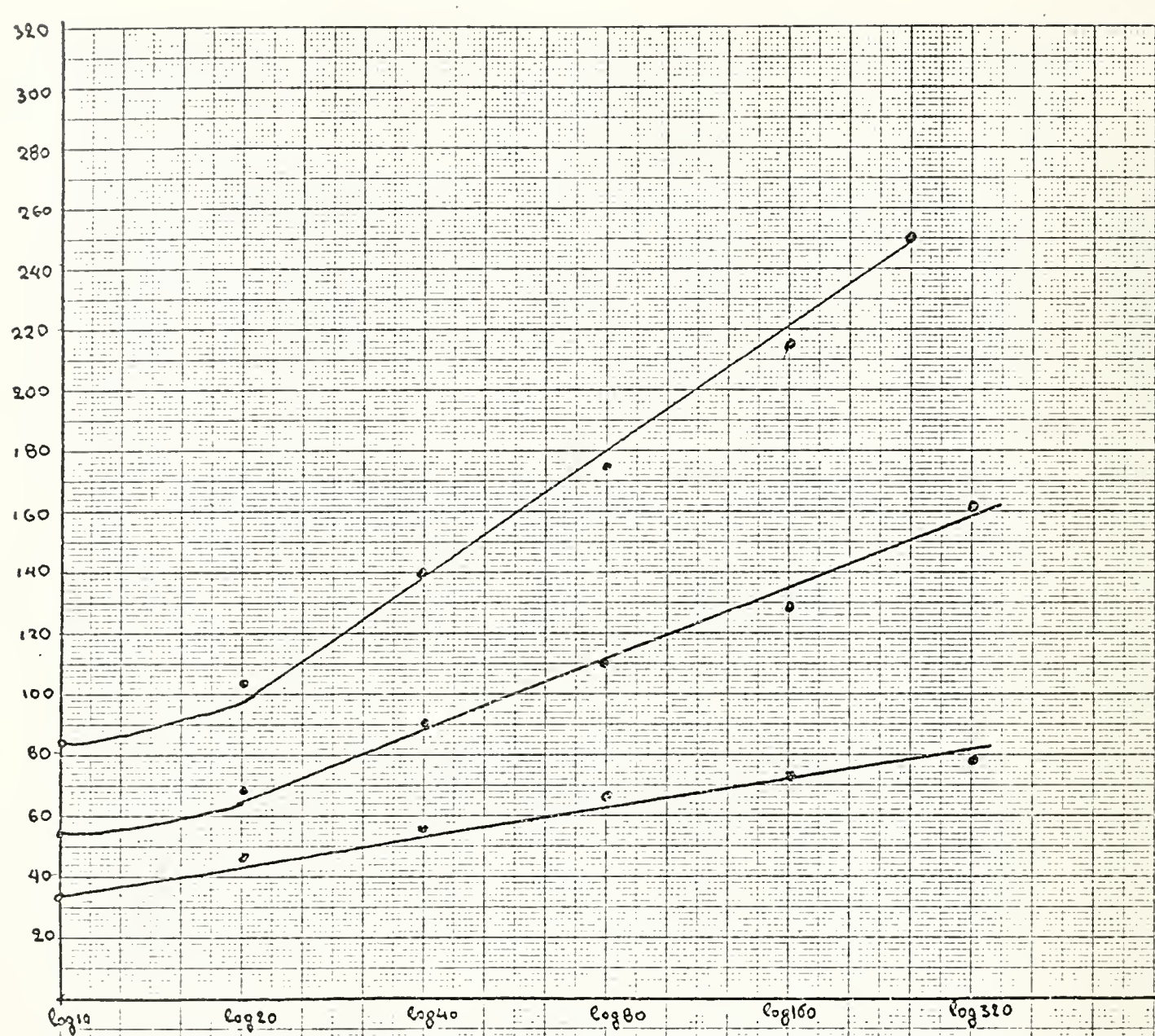


Figure 4.2.  $n_i$  versus  $\log(N)$  for cut = 100 (SD)

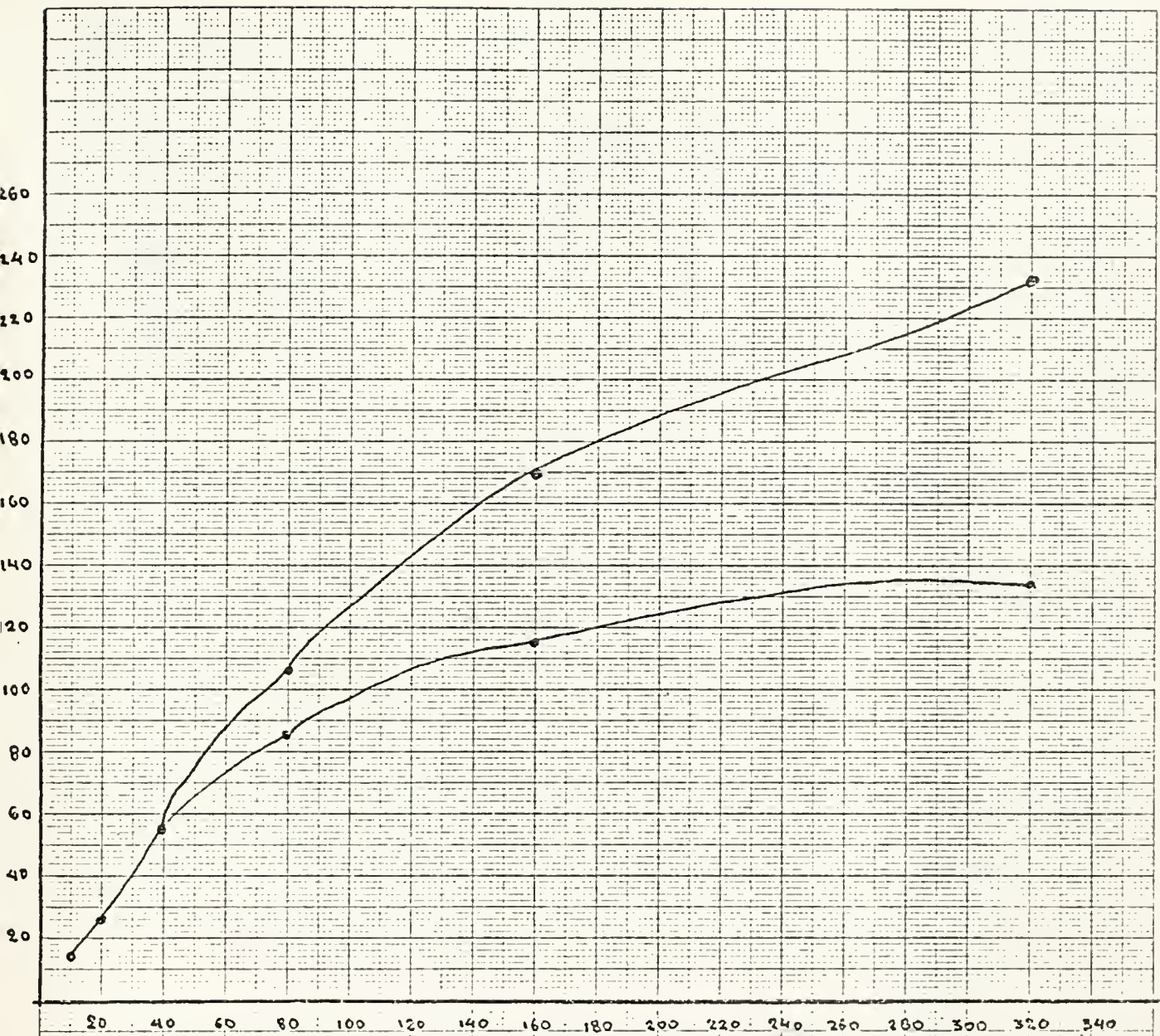


Figure 4.3.  $n_i$  versus  $N$  for  $\text{cut} = 4$  (SD)

A.1. Averages for measured quantities of  
the standard probabilistic (SP) system.

cut = 100

| ns  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 41 | 24 | 13 | 5  |     |     |
| 40  | 54 | 38 | 23 | 11 | 5   |     |
| 80  | 66 | 53 | 38 | 23 | 11  | 5   |
| 160 | 75 | 65 | 51 | 36 | 22  | 12  |
| 320 | 83 | 74 | 63 | 50 | 37  | 23  |

| nc  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 1  | 1  | 1  | 0  |     |     |
| 40  | 1  | 0  | 0  | 1  | 0   |     |
| 80  | 1  | 0  | 1  | 1  | 1   | 0   |
| 160 | 1  | 1  | 0  | 0  | 0   | 1   |
| 320 | 0  | 1  | 0  | 1  | 0   | 0   |

| nd  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 20 | 26 | 33 | 39 |     |     |
| 40  | 14 | 20 | 26 | 33 | 39  |     |
| 80  | 10 | 14 | 19 | 26 | 33  | 38  |
| 160 | 7  | 10 | 14 | 19 | 26  | 32  |
| 320 | 4  | 7  | 10 | 14 | 19  | 25  |

| ni  | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|-----|-----|-----|-----|
| 20  | 13 | 15 | 17  | 18  |     |     |
| 40  | 20 | 26 | 31  | 35  | 38  |     |
| 80  | 30 | 38 | 50  | 61  | 71  | 75  |
| 160 | 42 | 60 | 80  | 102 | 123 | 140 |
| 320 | 63 | 91 | 121 | 158 | 200 | 245 |

| nl  | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|-----|-----|-----|-----|
| 20  | 13 | 15 | 17  | 18  |     |     |
| 40  | 20 | 26 | 31  | 35  | 38  |     |
| 80  | 30 | 38 | 50  | 61  | 71  | 75  |
| 160 | 42 | 60 | 80  | 102 | 123 | 140 |
| 320 | 63 | 91 | 121 | 158 | 200 | 245 |

cut = 100 cont'd

| rw  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 7  | 16 | 36 | 76 |     |     |
| 40  | 6  | 16 | 36 | 76 | 155 |     |
| 80  | 6  | 16 | 35 | 75 | 154 | 311 |
| 160 | 6  | 16 | 35 | 75 | 153 | 308 |
| 320 | 6  | 15 | 35 | 74 | 152 | 310 |

| la  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 3  | 2  | 1  | 1  |     |     |
| 40  | 4  | 3  | 2  | 1  | 0   |     |
| 80  | 7  | 5  | 3  | 2  | 1   | 0   |
| 160 | 10 | 7  | 5  | 3  | 2   | 1   |
| 320 | 18 | 11 | 7  | 5  | 3   | 2   |

| wa  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 0  | 0  | 0  | 0  |     |     |
| 40  | 0  | 0  | 0  | 0  | 0   |     |
| 80  | 0  | 0  | 0  | 0  | 0   | 0   |
| 160 | 0  | 0  | 0  | 0  | 0   | 0   |
| 320 | 0  | 0  | 0  | 0  | 0   | 0   |

| m:la | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 8  | 6  | 4  | 3  |     |     |
| 40   | 11 | 8  | 6  | 4  | 3   |     |
| 80   | 16 | 11 | 8  | 6  | 4   | 3   |
| 160  | 21 | 16 | 10 | 8  | 6   | 4   |
| 320  | 30 | 21 | 15 | 12 | 8   | 6   |

| nact | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 2  | 2  | 1  | 1  |     |     |
| 40   | 2  | 2  | 2  | 2  | 1   |     |
| 80   | 2  | 3  | 3  | 2  | 2   | 2   |
| 160  | 3  | 3  | 3  | 3  | 3   | 4   |
| 320  | 3  | 4  | 4  | 4  | 5   | 4   |

| ns  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 44 | 26 | 13 | 6  |     |     |
| 40  | 57 | 40 | 26 | 13 | 5   |     |
| 80  | 68 | 54 | 39 | 26 | 13  | 6   |
| 160 | 78 | 67 | 54 | 39 | 25  | 14  |
| 320 | 85 | 76 | 65 | 54 | 39  | 24  |

| nc  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 6  | 6  | 6  | 6  |     |     |
| 40  | 5  | 5  | 6  | 6  | 6   |     |
| 80  | 6  | 6  | 5  | 6  | 6   | 5   |
| 160 | 6  | 6  | 5  | 6  | 5   | 6   |
| 320 | 6  | 6  | 6  | 6  | 6   | 6   |

| nd  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 17 | 24 | 31 | 37 |     |     |
| 40  | 12 | 17 | 24 | 30 | 37  |     |
| 80  | 8  | 12 | 17 | 23 | 30  | 36  |
| 160 | 5  | 7  | 12 | 17 | 23  | 29  |
| 320 | 3  | 5  | 7  | 11 | 16  | 23  |

| ni  | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|-----|-----|-----|-----|
| 20  | 12 | 15 | 17  | 18  |     |     |
| 40  | 18 | 24 | 28  | 34  | 37  |     |
| 80  | 27 | 38 | 49  | 60  | 70  | 74  |
| 160 | 40 | 54 | 73  | 97  | 120 | 136 |
| 320 | 51 | 82 | 114 | 150 | 193 | 240 |

| nl  | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|-----|-----|-----|-----|
| 20  | 12 | 15 | 17  | 18  |     |     |
| 40  | 18 | 24 | 28  | 34  | 37  |     |
| 80  | 27 | 38 | 49  | 60  | 70  | 74  |
| 160 | 40 | 54 | 73  | 97  | 120 | 136 |
| 320 | 51 | 82 | 114 | 150 | 193 | 240 |

cut = 16 cont'd

rw

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 6  | 16 | 36 | 76 |     |     |
| 40  | 6  | 15 | 36 | 76 | 155 |     |
| 80  | 6  | 14 | 35 | 75 | 154 | 310 |
| 160 | 5  | 15 | 34 | 74 | 153 | 308 |
| 320 | 5  | 15 | 33 | 73 | 152 | 310 |

la

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 3  | 2  | 1  | 1  |     |     |
| 40  | 4  | 3  | 2  | 1  | 1   |     |
| 80  | 7  | 5  | 3  | 2  | 1   | 0   |
| 160 | 10 | 7  | 5  | 3  | 2   | 1   |
| 320 | 13 | 9  | 7  | 5  | 3   | 2   |

wa

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 0  | 0  | 0  | 0  |     |     |
| 40  | 0  | 0  | 0  | 0  | 0   |     |
| 80  | 0  | 0  | 0  | 0  | 0   | 0   |
| 160 | 0  | 0  | 0  | 0  | 0   | 0   |
| 320 | 0  | 0  | 0  | 0  | 0   | 0   |

mxla

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 8  | 6  | 4  | 3  |     |     |
| 40  | 10 | 8  | 6  | 4  | 3   |     |
| 80  | 13 | 10 | 8  | 6  | 4   | 3   |
| 160 | 17 | 13 | 11 | 8  | 6   | 4   |
| 320 | 20 | 17 | 14 | 11 | 7   | 6   |

nact

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 2  | 2  | 1  | 1  |     |     |
| 40  | 2  | 3  | 2  | 2  | 1   |     |
| 80  | 3  | 4  | 3  | 3  | 2   | 3   |
| 160 | 3  | 4  | 4  | 3  | 3   | 4   |
| 320 | 3  | 3  | 5  | 4  | 5   | 5   |

| ns  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 45 | 27 | 14 | 5  |     |     |
| 40  | 58 | 42 | 27 | 13 | 5   |     |
| 80  | 70 | 56 | 41 | 26 | 13  | 6   |
| 160 | 79 | 69 | 54 | 40 | 26  | 14  |
| 320 | 86 | 77 | 66 | 54 | 40  | 24  |

| nc  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 8  | 8  | 8  | 7  |     |     |
| 40  | 7  | 8  | 7  | 8  | 8   |     |
| 80  | 8  | 8  | 8  | 8  | 7   | 7   |
| 160 | 8  | 9  | 8  | 7  | 8   | 7   |
| 320 | 8  | 7  | 8  | 8  | 7   | 8   |

| nd  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 16 | 23 | 30 | 37 |     |     |
| 40  | 11 | 16 | 23 | 30 | 36  |     |
| 80  | 6  | 11 | 16 | 22 | 30  | 35  |
| 160 | 4  | 6  | 11 | 16 | 22  | 29  |
| 320 | 2  | 4  | 7  | 10 | 15  | 22  |

| ni  | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|-----|-----|-----|-----|
| 20  | 11 | 15 | 17  | 18  |     |     |
| 40  | 17 | 24 | 29  | 35  | 37  |     |
| 80  | 25 | 36 | 47  | 59  | 69  | 75  |
| 160 | 37 | 51 | 73  | 97  | 120 | 137 |
| 320 | 50 | 74 | 103 | 147 | 193 | 240 |

| nl  | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|-----|-----|-----|-----|
| 20  | 11 | 15 | 17  | 18  |     |     |
| 40  | 17 | 24 | 29  | 35  | 37  |     |
| 80  | 25 | 36 | 47  | 59  | 69  | 75  |
| 160 | 37 | 51 | 73  | 97  | 120 | 137 |
| 320 | 50 | 74 | 103 | 147 | 193 | 240 |

cut = 12 cont'd

| rw  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 6  | 16 | 36 | 76 |     |     |
| 40  | 6  | 15 | 36 | 75 | 155 |     |
| 80  | 6  | 15 | 35 | 75 | 154 | 310 |
| 160 | 5  | 14 | 34 | 74 | 153 | 308 |
| 320 | 5  | 14 | 33 | 73 | 151 | 310 |

| la  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 3  | 2  | 1  | 1  |     |     |
| 40  | 4  | 3  | 2  | 1  | 0   |     |
| 80  | 7  | 4  | 3  | 2  | 1   | 0   |
| 160 | 9  | 7  | 4  | 3  | 2   | 1   |
| 320 | 13 | 10 | 6  | 4  | 3   | 2   |

| wa  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 0  | 0  | 0  | 0  |     |     |
| 40  | 0  | 0  | 0  | 0  | 0   |     |
| 80  | 0  | 0  | 0  | 0  | 0   | 0   |
| 160 | 0  | 0  | 0  | 0  | 0   | 0   |
| 320 | 0  | 0  | 0  | 0  | 0   | 0   |

| mxla | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 8  | 6  | 4  | 3  |     |     |
| 40   | 11 | 8  | 5  | 4  | 3   |     |
| 80   | 14 | 11 | 8  | 6  | 4   | 3   |
| 160  | 16 | 14 | 10 | 8  | 6   | 4   |
| 320  | 17 | 15 | 12 | 10 | 8   | 6   |

| nact | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 2  | 2  | 1  | 1  |     |     |
| 40   | 2  | 3  | 2  | 2  | 2   |     |
| 80   | 3  | 3  | 3  | 2  | 2   | 2   |
| 160  | 3  | 4  | 4  | 4  | 3   | 4   |
| 320  | 3  | 5  | 5  | 5  | 5   | 5   |

cut = 8

| ns  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 46 | 28 | 15 | 6  |     |     |
| 40  | 59 | 44 | 27 | 14 | 7   |     |
| 80  | 72 | 57 | 44 | 27 | 14  | 7   |
| 160 | 81 | 69 | 57 | 42 | 27  | 14  |
| 320 | 88 | 80 | 70 | 56 | 42  | 27  |

| nc  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 12 | 12 | 12 | 12 |     |     |
| 40  | 12 | 12 | 12 | 12 | 12  |     |
| 80  | 13 | 13 | 12 | 12 | 12  | 11  |
| 160 | 12 | 12 | 12 | 12 | 12  | 12  |
| 320 | 12 | 12 | 12 | 12 | 12  | 12  |

| nd  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 15 | 21 | 28 | 35 |     |     |
| 40  | 10 | 15 | 21 | 28 | 34  |     |
| 80  | 5  | 9  | 13 | 21 | 27  | 33  |
| 160 | 3  | 6  | 9  | 14 | 20  | 27  |
| 320 | 1  | 2  | 5  | 9  | 14  | 20  |

| ni  | 10 | 20 | 40 | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|
| 20  | 11 | 14 | 16 | 18  |     |     |
| 40  | 16 | 22 | 29 | 34  | 37  |     |
| 80  | 25 | 35 | 46 | 58  | 68  | 74  |
| 160 | 35 | 50 | 70 | 92  | 117 | 136 |
| 320 | 42 | 66 | 98 | 141 | 183 | 234 |

| nl  | 10 | 20 | 40 | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|
| 20  | 11 | 14 | 16 | 18  |     |     |
| 40  | 16 | 22 | 29 | 34  | 37  |     |
| 80  | 25 | 35 | 46 | 58  | 68  | 74  |
| 160 | 35 | 50 | 70 | 92  | 117 | 136 |
| 320 | 42 | 66 | 98 | 141 | 183 | 234 |

cut = 8 cont'd

nw

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 6  | 15 | 36 | 76 |     |     |
| 40  | 5  | 15 | 35 | 75 | 155 |     |
| 80  | 5  | 14 | 34 | 75 | 154 | 310 |
| 160 | 5  | 14 | 34 | 73 | 153 | 307 |
| 320 | 4  | 13 | 32 | 72 | 151 | 309 |

la

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 3  | 2  | 1  | 1  |     |     |
| 40  | 4  | 3  | 2  | 1  | 1   |     |
| 80  | 6  | 4  | 3  | 2  | 1   | 0   |
| 160 | 9  | 6  | 4  | 3  | 2   | 1   |
| 320 | 9  | 9  | 6  | 4  | 3   | 2   |

wa

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 0  | 0  | 0  | 0  |     |     |
| 40  | 0  | 0  | 0  | 0  | 0   |     |
| 80  | 0  | 0  | 0  | 0  | 0   | 0   |
| 160 | 0  | 0  | 0  | 0  | 0   | 0   |
| 320 | 0  | 0  | 0  | 0  | 0   | 0   |

mxla

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 7  | 5  | 4  | 3  |     |     |
| 40  | 10 | 7  | 5  | 4  | 3   |     |
| 80  | 11 | 9  | 8  | 6  | 4   | 3   |
| 160 | 13 | 12 | 10 | 7  | 6   | 4   |
| 320 | 11 | 13 | 12 | 10 | 7   | 5   |

nact

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 2  | 2  | 2  | 1  |     |     |
| 40  | 3  | 3  | 2  | 2  | 1   |     |
| 80  | 3  | 3  | 3  | 2  | 2   | 3   |
| 160 | 3  | 4  | 4  | 4  | 3   | 4   |
| 320 | 4  | 5  | 5  | 5  | 5   | 5   |

| ris | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 53 | 33 | 18 | 8  |     |     |
| 40  | 68 | 50 | 33 | 17 | 8   |     |
| 80  | 79 | 64 | 49 | 33 | 18  | 8   |
| 160 | 87 | 77 | 64 | 49 | 32  | 18  |
| 320 | 93 | 85 | 77 | 62 | 47  | 30  |

| ric | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 25 | 24 | 24 | 24 |     |     |
| 40  | 25 | 24 | 24 | 25 | 24  |     |
| 80  | 24 | 24 | 26 | 24 | 25  | 25  |
| 160 | 25 | 25 | 24 | 24 | 25  | 24  |
| 320 | 25 | 25 | 24 | 25 | 24  | 24  |

| rid | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 10 | 16 | 23 | 29 |     |     |
| 40  | 5  | 10 | 15 | 22 | 29  |     |
| 80  | 2  | 5  | 9  | 15 | 22  | 28  |
| 160 | 1  | 2  | 5  | 9  | 15  | 21  |
| 320 | 0  | 0  | 2  | 5  | 9   | 15  |

| ri  | 10 | 20 | 40 | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|
| 20  | 10 | 13 | 16 | 18  |     |     |
| 40  | 14 | 20 | 27 | 33  | 36  |     |
| 80  | 17 | 29 | 41 | 54  | 65  | 73  |
| 160 | 24 | 37 | 58 | 84  | 109 | 131 |
| 320 | 27 | 46 | 76 | 118 | 167 | 223 |

| ri  | 10 | 20 | 40 | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|
| 20  | 10 | 13 | 16 | 18  |     |     |
| 40  | 14 | 20 | 27 | 33  | 36  |     |
| 80  | 17 | 29 | 41 | 54  | 65  | 73  |
| 160 | 24 | 37 | 58 | 84  | 109 | 131 |
| 320 | 27 | 46 | 76 | 118 | 167 | 223 |

cut = 4 cont'd

| rw  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 5  | 15 | 34 | 75 |     |     |
| 40  | 4  | 14 | 34 | 74 | 154 |     |
| 80  | 3  | 12 | 33 | 73 | 153 | 308 |
| 160 | 3  | 11 | 31 | 71 | 151 | 306 |
| 320 | 2  | 9  | 27 | 68 | 146 | 307 |

| la  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 2  | 2  | 1  | 0  |     |     |
| 40  | 3  | 2  | 2  | 1  | 0   |     |
| 80  | 4  | 4  | 2  | 2  | 1   | 0   |
| 160 | 4  | 5  | 4  | 3  | 2   | 1   |
| 320 | 1  | 3  | 4  | 4  | 3   | 2   |

| wa  | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 20  | 0  | 0  | 0  | 0  |     |     |
| 40  | 0  | 0  | 0  | 0  | 0   |     |
| 80  | 0  | 0  | 0  | 0  | 0   | 0   |
| 160 | 0  | 0  | 0  | 0  | 0   | 0   |
| 320 | 0  | 0  | 0  | 0  | 0   | 0   |

| m×la | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 6  | 5  | 4  | 3  |     |     |
| 40   | 7  | 6  | 5  | 4  | 3   |     |
| 80   | 6  | 8  | 6  | 5  | 4   | 3   |
| 160  | 5  | 7  | 8  | 6  | 5   | 4   |
| 320  | 1  | 4  | 7  | 8  | 6   | 5   |

| nact | 10 | 20 | 40 | 80 | 160 | 320 |
|------|----|----|----|----|-----|-----|
| 20   | 2  | 2  | 2  | 1  |     |     |
| 40   | 3  | 3  | 3  | 2  | 2   |     |
| 80   | 4  | 4  | 4  | 3  | 2   | 3   |
| 160  | 4  | 6  | 6  | 5  | 4   | 4   |
| 320  | 5  | 7  | 8  | 7  | 7   | 6   |

## A.2 Data and trends for the parallel deterministic (PD) system.

( a )     cut = 100

No completions were recorded.

ni: increases from  $\log (N/M)$  to  $M$ .

nd: proportional to  $\log (N/M)$ .

ns: increasing almost like  $-\log (N/M)$  and then flattening.

mxla: proportional to  $\log (N/M)$ .

nw: proportional to  $N$  (and very large).

cut = 100

ns

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 10  |    |    |    |    |     |     |
| 20  | 35 | 20 | 7  | 1  |     |     |
| 40  | 46 | 31 | 17 | 6  | 1   |     |
| 80  | 58 | 43 | 29 | 16 | 6   | 2   |
| 160 | 68 | 58 | 43 | 28 | 15  | 7   |
| 320 | 78 | 68 | 55 | 41 | 28  | 15  |

nd

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 10  |    |    |    |    |     |     |
| 20  | 15 | 18 | 20 | 22 |     |     |
| 40  | 11 | 14 | 17 | 19 | 21  |     |
| 80  | 8  | 11 | 13 | 16 | 18  | 19  |
| 160 | 6  | 8  | 10 | 12 | 15  | 17  |
| 320 | 4  | 5  | 7  | 9  | 12  | 15  |

ni

|     | 10  | 20  | 40  | 80  | 160 | 320 |
|-----|-----|-----|-----|-----|-----|-----|
| 10  |     |     |     |     |     |     |
| 20  | 17  | 19  | 19  | 19  |     |     |
| 40  | 29  | 34  | 37  | 39  | 39  |     |
| 80  | 48  | 59  | 69  | 76  | 79  | 79  |
| 160 | 74  | 93  | 117 | 139 | 151 | 157 |
| 320 | 107 | 147 | 191 | 235 | 274 | 303 |

nw

|     | 10 | 20 | 40 | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|
| 10  |    |    |    |     |     |     |
| 20  | 13 | 26 | 60 | 133 |     |     |
| 40  | 14 | 28 | 57 | 133 | 279 |     |
| 80  | 13 | 30 | 67 | 126 | 269 | 569 |
| 160 | 14 | 29 | 66 | 131 | 271 | 544 |
| 320 | 11 | 28 | 66 | 133 | 276 | 575 |

mxla

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 10  |    |    |    |    |     |     |
| 20  | 10 | 7  | 5  | 2  |     |     |
| 40  | 14 | 9  | 7  | 5  | 3   |     |
| 80  | 20 | 13 | 10 | 7  | 5   | 3   |
| 160 | 24 | 17 | 13 | 9  | 6   | 5   |
| 320 | 31 | 23 | 17 | 13 | 10  | 7   |

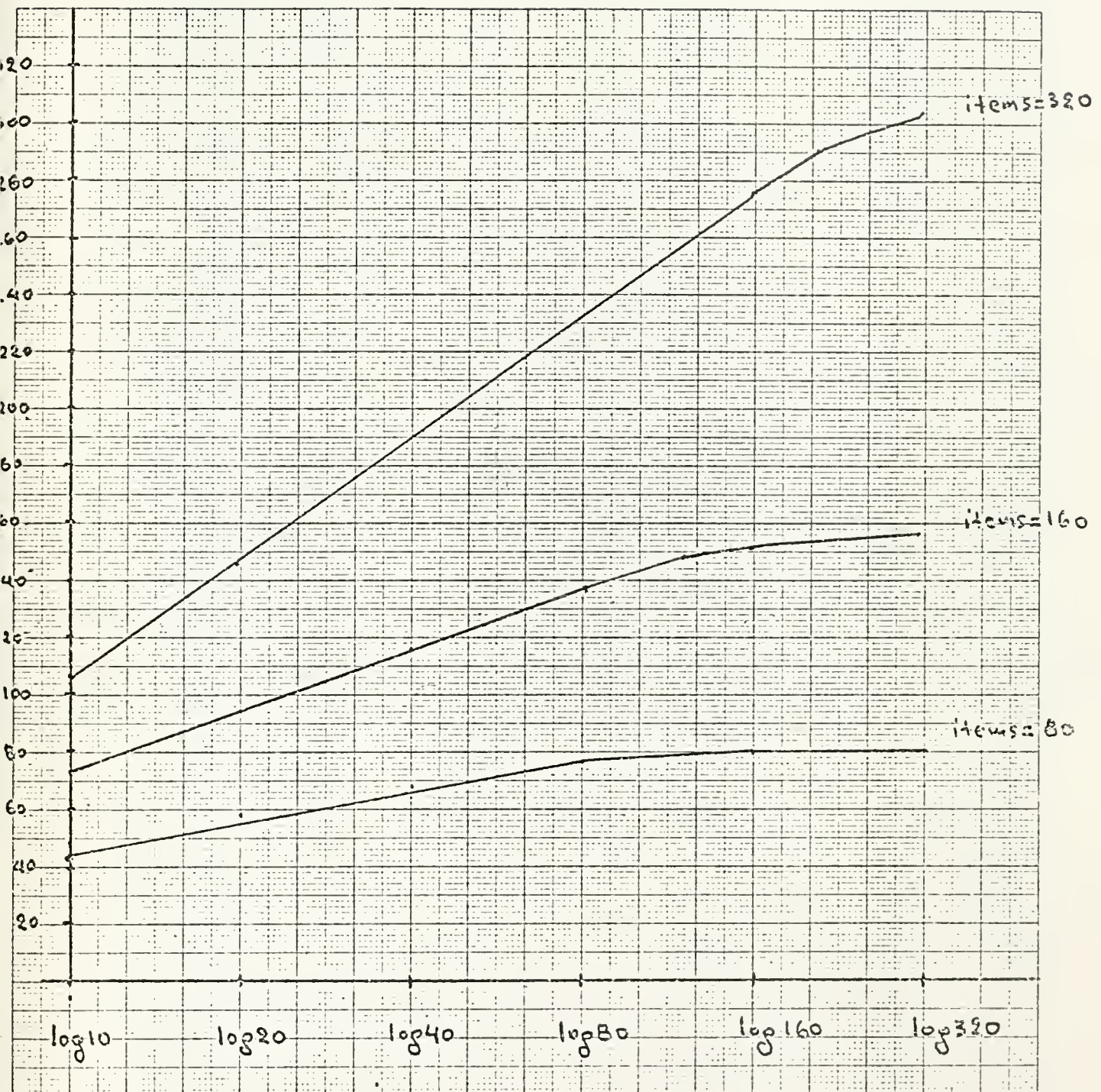


Figure 8.1

Items locked vs log N

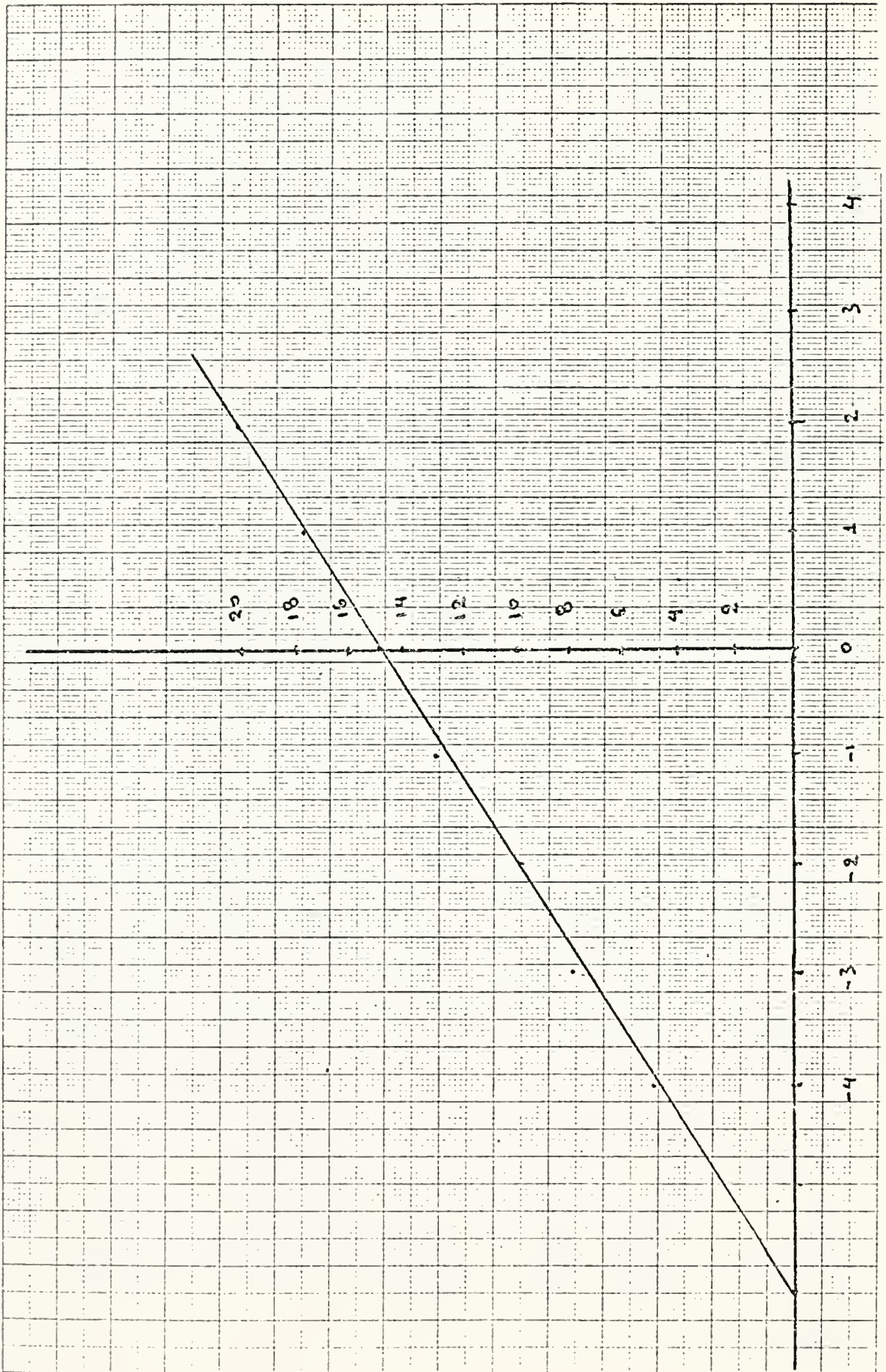
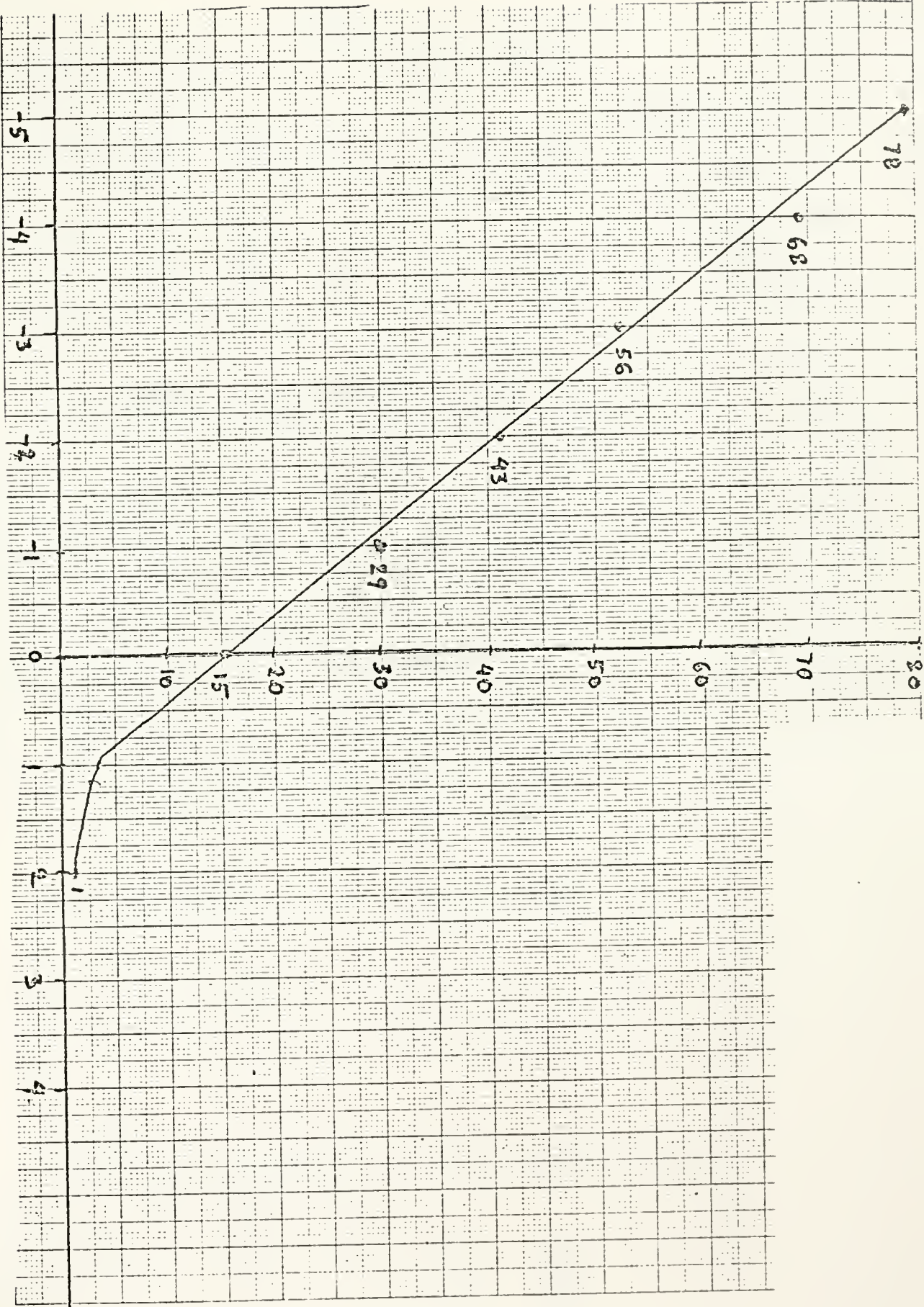
Figure 8.2 Deadlock rate (nd/100 requests) vs  $\log(N/M)$

Figure 8.3 ns per 100 lock requests versus  $\log_2(N/M)$ 

(b) cut = 4

ni: proportional to  $N$

nd: proportional to  $\log N/M$

mxla: negligible

ns: no conclusion

nw: proportional to  $N^{\log_2 3}$

number of completions after threshold: proportional to  $\log N/M$

cut = 4

ns

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 10  |    |    |    |    |     |     |
| 20  | 58 | 29 | 10 | 2  |     |     |
| 40  | 76 | 54 | 26 | 8  | 1   |     |
| 80  | 87 | 72 | 50 | 23 | 8   | 2   |
| 160 | 93 | 86 | 71 | 47 | 22  | 7   |
| 320 | 96 | 93 | 85 | 71 | 46  | 21  |

nc

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 10  |    |    |    |    |     |     |
| 20  | 18 | 12 | 7  | 2  |     |     |
| 40  | 22 | 19 | 14 | 8  | 4   |     |
| 80  | 24 | 23 | 20 | 15 | 9   | 4   |
| 160 | 24 | 24 | 23 | 21 | 15  | 9   |
| 320 | 24 | 24 | 24 | 24 | 21  | 17  |

ni

|     | 10 | 20 | 40 | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|
| 10  |    |    |    |     |     |     |
| 20  | 11 | 17 | 19 | 20  |     |     |
| 40  | 14 | 24 | 34 | 38  | 39  |     |
| 80  | 14 | 29 | 50 | 70  | 77  | 79  |
| 160 | 15 | 29 | 58 | 105 | 140 | 156 |
| 320 | 15 | 30 | 60 | 120 | 208 | 284 |

nd

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 10  |    |    |    |    |     |     |
| 20  | 6  | 13 | 19 | 24 |     |     |
| 40  | 2  | 5  | 10 | 17 | 21  |     |
| 80  | 0  | 1  | 4  | 9  | 14  | 20  |
| 160 | 0  | 0  | 3  | 8  | 14  | 20  |
| 320 | 0  | 0  | 0  | 0  | 2   | 6   |

nw

|     | 10 | 20 | 40 | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|
| 10  |    |    |    |     |     |     |
| 20  | 4  | 19 | 55 | 131 |     |     |
| 40  | 3  | 13 | 42 | 116 | 285 |     |
| 80  | 1  | 8  | 31 | 98  | 256 | 596 |
| 160 | 1  | 4  | 18 | 71  | 206 | 543 |
| 320 | 0  | 2  | 9  | 39  | 143 | 443 |

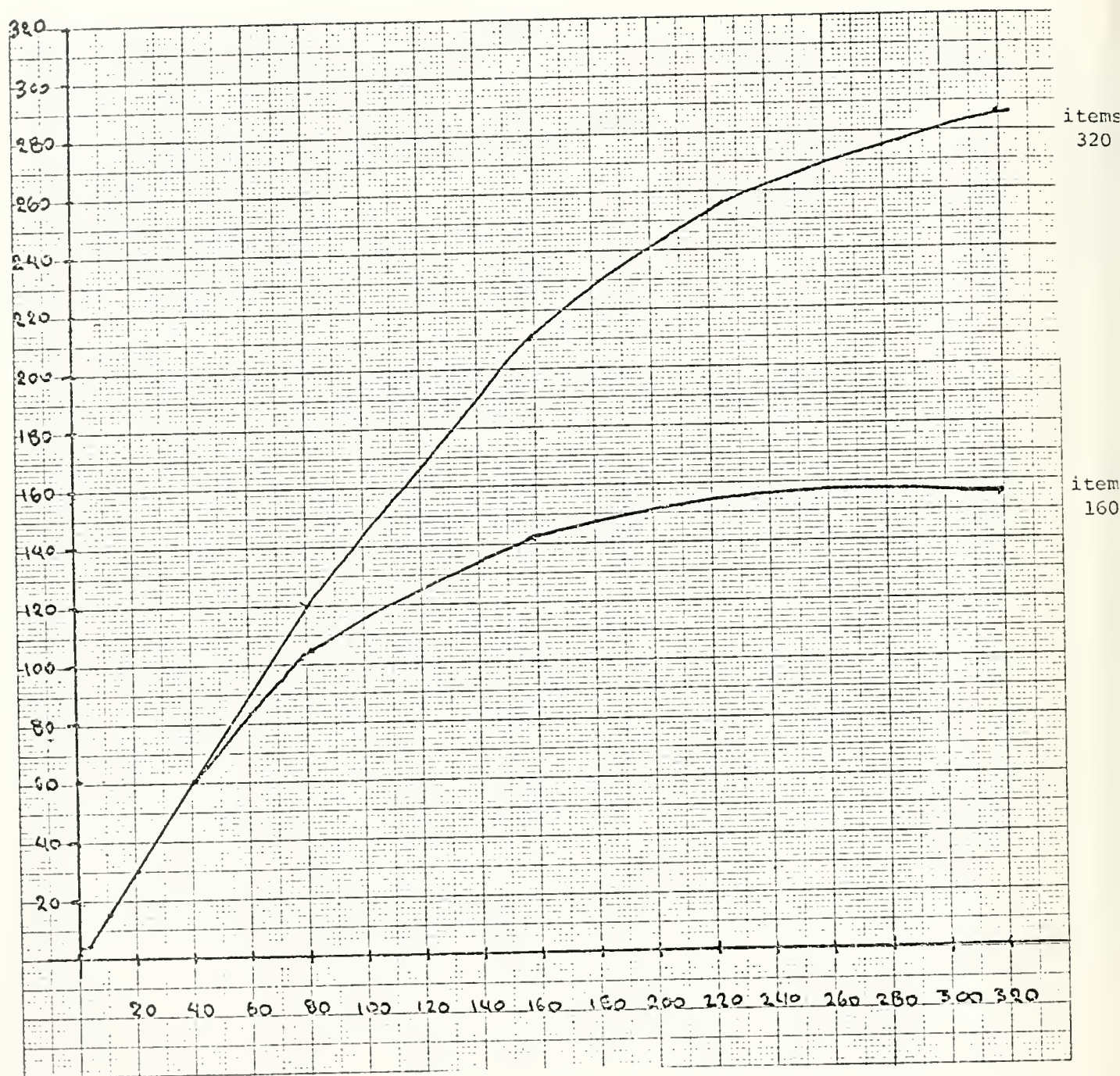


Figure 8.4. Number of locked items ( $n_i$ ) versus  $N$

(c) cut = 16

Only a few completions. Very similar to cut = 100.

cut = 16

ni

|     | 10 | 20  | 40  | 80  | 160 | 320 |
|-----|----|-----|-----|-----|-----|-----|
| 10  |    |     |     |     |     |     |
| 20  | 16 | 18  | 19  | 19  |     |     |
| 40  | 28 | 34  | 37  | 39  | 39  |     |
| 80  | 45 | 58  | 68  | 76  | 79  | 79  |
| 160 | 56 | 92  | 117 | 138 | 152 | 157 |
| 320 | 71 | 121 | 187 | 240 | 273 | 304 |

nd

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 10  |    |    |    |    |     |     |
| 20  | 15 | 18 | 21 | 22 |     |     |
| 40  | 11 | 14 | 17 | 19 | 20  |     |
| 80  | 7  | 10 | 13 | 15 | 18  | 19  |
| 160 | 4  | 7  | 10 | 13 | 15  | 18  |
| 320 | 2  | 4  | 6  | 9  | 12  | 15  |

(d) cut = 8

ni: slower than linear to  $N$

nd: proportional to  $\log N/M$

ns: no remarks

mxla: decreases with  $N$

nw: proportional to  $N^{\log_2(3\delta)}$  for  $\delta \in [1, 2]$

number of completions: proportional to  $\log(N/M)$  (after a threshold).

cut = 8

ns

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 10  |    |    |    |    |     |     |
| 20  | 37 | 19 | 7  | 1  |     |     |
| 40  | 55 | 35 | 18 | 6  | 1   |     |
| 80  | 73 | 51 | 31 | 15 | 5   | 1   |
| 160 | 85 | 70 | 50 | 29 | 13  | 6   |
| 320 | 92 | 84 | 69 | 46 | 26  | 13  |

nc

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 10  |    |    |    |    |     |     |
| 20  | 2  | 0  | 0  | 0  |     |     |
| 40  | 5  | 2  | 0  | 0  | 0   |     |
| 80  | 8  | 5  | 2  | 1  | 0   | 0   |
| 160 | 11 | 9  | 6  | 3  | 0   | 0   |
| 320 | 12 | 11 | 9  | 7  | 3   | 1   |

ni

|     | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|-----|-----|-----|-----|
| 10  |    |    |     |     |     |     |
| 20  | 16 | 19 | 19  | 19  |     |     |
| 40  | 23 | 33 | 37  | 39  | 39  |     |
| 80  | 30 | 50 | 68  | 76  | 79  | 79  |
| 160 | 32 | 63 | 102 | 136 | 152 | 157 |
| 320 | 36 | 72 | 138 | 218 | 271 | 305 |

nd

|     | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|----|----|----|----|-----|-----|
| 10  |    |    |    |    |     |     |
| 20  | 14 | 18 | 20 | 22 |     |     |
| 40  | 8  | 13 | 17 | 19 | 21  |     |
| 80  | 4  | 8  | 12 | 15 | 18  | 19  |
| 160 | 1  | 3  | 6  | 11 | 14  | 17  |
| 320 | 0  | 0  | 2  | 5  | 9   | 14  |

nw

|     | 10 | 20 | 40 | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|
| 10  |    |    |    |     |     |     |
| 20  | 10 | 29 | 60 | 135 |     |     |
| 40  | 8  | 24 | 61 | 136 | 285 |     |
| 80  | 6  | 22 | 61 | 137 | 281 | 578 |
| 160 | 3  | 14 | 48 | 130 | 314 | 589 |
| 320 | 2  | 8  | 37 | 116 | 306 | 638 |

### A.3 The Analytic Model of Balls and Urns for SP.

All tables refer to nw.

$$\theta = 1/100$$

|     | 10 | 20  | 40  | 80  | 160 | 320 |
|-----|----|-----|-----|-----|-----|-----|
| 10  | 8  | 9   | 10  | 10  | 10  | 10  |
| 20  | 14 | 17  | 19  | 20  | 20  | 20  |
| 40  | 22 | 28  | 34  | 38  | 39  | 40  |
| 80  | 34 | 44  | 56  | 67  | 76  | 79  |
| 160 | 50 | 68  | 88  | 112 | 135 | 152 |
| 320 | 74 | 101 | 135 | 177 | 224 | 269 |

$$\theta = 1/16$$

|     | 5  | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|-----|
| 10  | 7  | 8  | 10 | 10  | 10  | 10  | 10  |
| 20  | 11 | 14 | 17 | 19  | 20  | 20  | 20  |
| 40  | 16 | 21 | 27 | 33  | 38  | 40  | 40  |
| 80  | 22 | 31 | 42 | 55  | 67  | 76  | 79  |
| 160 | 30 | 44 | 62 | 84  | 109 | 133 | 151 |
| 320 | 39 | 60 | 87 | 123 | 168 | 218 | 266 |

$$\theta = 1/12$$

|     | 5  | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|-----|
| 10  | 7  | 8  | 9  | 10  | 10  | 10  | 10  |
| 20  | 11 | 14 | 17 | 19  | 20  | 20  | 20  |
| 40  | 15 | 21 | 27 | 33  | 38  | 40  | 40  |
| 80  | 21 | 30 | 41 | 54  | 66  | 76  | 79  |
| 160 | 28 | 42 | 60 | 82  | 108 | 133 | 150 |
| 320 | 35 | 56 | 83 | 120 | 165 | 215 | 265 |

$$\theta = 1/8$$

|     | 5  | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|-----|
| 10  | 7  | 8  | 9  | 10  | 10  | 10  | 10  |
| 20  | 10 | 13 | 17 | 19  | 20  | 20  | 20  |
| 40  | 14 | 20 | 27 | 33  | 38  | 40  | 40  |
| 80  | 19 | 28 | 40 | 53  | 66  | 75  | 79  |
| 160 | 24 | 38 | 56 | 79  | 106 | 131 | 151 |
| 320 | 29 | 48 | 76 | 130 | 159 | 211 | 263 |

$$\theta = 1/6$$

|     | 5  | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|-----|
| 10  | 7  | 8  | 9  | 10  | 10  | 10  | 10  |
| 20  | 10 | 13 | 16 | 19  | 20  | 20  | 20  |
| 40  | 13 | 19 | 26 | 33  | 38  | 40  | 40  |
| 80  | 17 | 27 | 38 | 52  | 65  | 75  | 79  |
| 160 | 21 | 35 | 53 | 77  | 104 | 130 | 150 |
| 320 | 24 | 42 | 69 | 106 | 153 | 207 | 260 |

$$\theta = 1/4$$

|     | 5  | 10 | 20 | 40 | 80  | 160 | 320 |
|-----|----|----|----|----|-----|-----|-----|
| 10  | 6  | 8  | 9  | 10 | 10  | 10  | 10  |
| 20  | 9  | 13 | 16 | 19 | 20  | 20  | 20  |
| 40  | 12 | 18 | 25 | 32 | 37  | 40  | 40  |
| 80  | 14 | 23 | 35 | 50 | 64  | 75  | 79  |
| 160 | 17 | 29 | 47 | 71 | 99  | 128 | 149 |
| 320 | 18 | 33 | 57 | 93 | 141 | 198 | 255 |

$$\theta = 1/2$$

|     | 5  | 10 | 20 | 40 | 80  | 160 | 320 |
|-----|----|----|----|----|-----|-----|-----|
| 10  | 5  | 8  | 9  | 10 | 10  | 10  | 10  |
| 20  | 7  | 11 | 15 | 18 | 20  | 20  | 20  |
| 40  | 8  | 14 | 22 | 30 | 36  | 40  | 40  |
| 80  | 9  | 16 | 28 | 43 | 60  | 73  | 79  |
| 160 | 9  | 18 | 32 | 55 | 86  | 120 | 146 |
| 320 | 10 | 19 | 36 | 65 | 110 | 171 | 239 |

$$\theta = 1/10$$

|     | 5 | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|---|----|----|----|----|-----|-----|
| 10  | 4 | 7  | 9  | 10 | 10 | 10  | 10  |
| 20  | 5 | 8  | 13 | 17 | 20 | 20  | 20  |
| 40  | 5 | 9  | 16 | 25 | 35 | 40  | 40  |
| 80  | 5 | 9  | 18 | 32 | 51 | 70  | 79  |
| 160 | 5 | 10 | 19 | 35 | 63 | 101 | 138 |
| 320 | 5 | 10 | 19 | 38 | 71 | 126 | 202 |

$$\theta = 1$$

|     | 5  | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|-----|
| 10  | 7  | 8  | 9  | 10  | 10  | 10  | 10  |
| 20  | 10 | 14 | 17 | 19  | 20  | 20  | 20  |
| 40  | 15 | 21 | 27 | 33  | 38  | 40  | 40  |
| 80  | 20 | 29 | 41 | 54  | 66  | 75  | 79  |
| 160 | 26 | 40 | 58 | 81  | 107 | 132 | 151 |
| 320 | 33 | 52 | 80 | 117 | 162 | 213 | 264 |

( $\theta = 1$  is just the classical occupancy problem)

$$\theta = 1/10$$

|     | 5 | 10 | 20 | 40 | 80 | 160 | 320 |
|-----|---|----|----|----|----|-----|-----|
| 10  | 4 | 7  | 9  | 10 | 10 | 10  | 10  |
| 20  | 5 | 8  | 13 | 17 | 20 | 20  | 20  |
| 40  | 5 | 9  | 16 | 25 | 35 | 40  | 40  |
| 80  | 5 | 9  | 18 | 32 | 51 | 70  | 79  |
| 160 | 5 | 10 | 19 | 35 | 63 | 101 | 138 |
| 320 | 5 | 10 | 19 | 38 | 71 | 126 | 202 |

$$\theta = 1$$

|     | 5  | 10 | 20 | 40  | 80  | 160 | 320 |
|-----|----|----|----|-----|-----|-----|-----|
| 10  | 7  | 8  | 9  | 10  | 10  | 10  | 10  |
| 20  | 10 | 14 | 17 | 19  | 20  | 20  | 20  |
| 40  | 15 | 21 | 27 | 33  | 38  | 40  | 40  |
| 80  | 20 | 29 | 41 | 54  | 66  | 75  | 79  |
| 160 | 26 | 40 | 58 | 81  | 107 | 132 | 151 |
| 320 | 33 | 52 | 80 | 117 | 162 | 213 | 264 |

( $\theta = 1$  is just the classical occupancy problem)

NOV 1 1983

FOURTEEN DAYS

A fine will be charged for each day the book is kept overtime.

|             |  |  |                  |
|-------------|--|--|------------------|
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
|             |  |  |                  |
| GAYLORD 142 |  |  | PRINTED IN U S A |

NYU c.2  
Comp. Sci. Dept.  
TR-54 Shmueli  
Concurrency control per-  
formance evaluation (a...

|                            |                 |
|----------------------------|-----------------|
| NYU c.2                    |                 |
| Comp. Sci. Dept.           |                 |
| TR-54 Shmueli              |                 |
| AUTHOR                     |                 |
| Concurrency control per-   |                 |
| TITLE                      |                 |
| formance evaluation (a ... |                 |
| DATE DUE                   | BORROWER'S NAME |
|                            |                 |
|                            |                 |
|                            |                 |
|                            |                 |

**N.Y.U. Courant Institute of  
Mathematical Sciences**

**251 Mercer St.  
New York, N. Y. 10012**

